

NAVAL POSTGRADUATE SCHOOL

Monterey, California



THESIS

THE USE OF CONFORMAL SUBDOMAIN
BASIS FUNCTIONS IN THE METHOD OF MOMENTS
COMPUTATIONS FOR A THIN WIRE

by

Bruce A. Walter

December, 1991

Thesis Advisor:

David C. Jenn

Approved for public release; distribution is unlimited.

T259071

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED			1b RESTRICTIVE MARKINGS NONE		
2a SECURITY CLASSIFICATION AUTHORITY N/A			3 DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.		
2b DECLASSIFICATION / DOWNGRADING SCHEDULE					
4. PERFORMING ORGANIZATION REPORT NUMBER(S)			5 MONITORING ORGANIZATION REPORT NUMBER(S)		
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School		6b OFFICE SYMBOL (If applicable) 3A	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School		
6c. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000			7b. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000		
8a NAME OF FUNDING / SPONSORING ORGANIZATION		8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER		
8c. ADDRESS (City, State, and ZIP Code)			10 SOURCE OF FUNDING NUMBERS		
			PROGRAM ELEMENT NO	PROJECT NO	TASK NO
			WORK UNIT ACCESSION NO		
11 TITLE (Include Security Classification) THE USE OF CONFORMAL SUBDOMAIN BASIS FUNCTIONS IN THE METHOD OF MOMENTS COMPUTATIONS FOR A THIN WIRE					
12 PERSONAL AUTHOR(S) Walter, Bruce A.					
13a TYPE OF REPORT Master's Thesis		13b TIME COVERED FROM _____ TO _____		14 DATE OF REPORT (Year, Month, Day) 1991, December	
				15 PAGE COUNT 90	
16 SUPPLEMENTARY NOTATION The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U. S. Government.					
17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB-GROUP	Method of Moments Conformal Subdomain Basis Functions Thin Wire Integral Equation		
19 ABSTRACT (Continue on reverse if necessary and identify by block number)					
<p>The purpose of this thesis is to investigate the use of Conformal Subdomain Basis Functions (CSBF) in the Method of Moments (MM) solution of a thin wire scatterer. The effect of using CSBF on the computed current and the scattered field is investigated by formulating and coding the MM solution for a thin wire loop and comparing the computed results for various loop sizes to measured data and two other MM codes. Significant reduction in the number of segments (and computer memory requirements) are found for loops with circumferences of less than one to two wavelengths for plane wave incidence. From these results, it is concluded that the use of CSBF will significantly reduce the number of segments required for the MM solution of a spiral antenna.</p>					
20 DISTRIBUTION / AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS			21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED		
22a NAME OF RESPONSIBLE INDIVIDUAL D. C. Jenn			22b TELEPHONE (Include Area Code) 408-646-2254		22c OFFICE SYMBOL CODE EC/Jn

Approved for public release; distribution is unlimited.

THE USE OF CONFORMAL SUBDOMAIN
BASIS FUNCTIONS IN THE METHOD OF MOMENTS
COMPUTATIONS FOR A THIN WIRE

by

Bruce A. Walter
B.S.E.E., Virginia Polytechnic Institute, 1984

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December, 1991

Department of Electrical and Computer Engineering

ABSTRACT

The purpose of this thesis is to investigate the use of Conformal Subdomain Basis Functions (CSBF) in the Method of Moments (MM) solution of a thin wire scatterer. The effect of using CSBF on the computed current and the scattered field is investigated by formulating and coding the MM solution for a thin wire loop and comparing the computed results for various loop sizes to measured data and two other MM codes. Significant reduction in the number of segments (and computer memory requirements) are found for loops with circumferences of less than one to two wavelengths for plane wave incidence. From these results, it is concluded that the use of CSBF will significantly reduce the number of segments required for the MM solution of a spiral antenna.

T Res 13
U, 22415
c.1

TABLE OF CONTENTS

I. INTRODUCTION	1
II. THE THIN WIRE INTEGRAL EQUATION	4
A. DERIVATION OF THE THIN-WIRE ELECTRIC FIELD INTEGRAL EQUATION	4
B. SOLUTION OF THE EFIE USING MM	7
C. SPECIALIZATION OF THE EFIE TO A CIRCULAR LOOP USING CONFORMAL SUBSECTIONS.	10
III. COMPUTER CODES FOR THE THIN WIRE LOOP	16
A. DESCRIPTION OF THE CODES	16
1. Loop Geometry	18
2. Subroutines ZCURVED and ZMATWW	18
3. Execution Time	21
IV. CALCULATED DATA FOR THE LOOP	24
A. CONVERGENCE OF HARLOOP	24
B. CONVERGENCE OF THE CURRENT EXPANSION	26

C. COMPARISON OF EXECUTION TIME AND MEMORY REQUIREMENTS	37
V. CONCLUSIONS	39
REFERENCES	43
APPENDIX A. COMPUTER CODES	44
APPENDIX B. ADDITIONAL PLOTS OF CURRENT AND RMS ERROR . . .	74
INITIAL DISTRIBUTION LIST	83

I. INTRODUCTION

Numerical techniques for solving electromagnetic scattering problems using integral equations and the method of moments (MM) are well known. The physical problem, specified by Maxwell's equations and boundary conditions, is reduced to an integro-differential equation over finite domains, and solved using a procedure referred to as the method of moments [Ref. 1]. The unknowns (usually currents) are represented by a series of basis functions with unknown expansion coefficients. The MM process generates a set of linear equations that must be solved simultaneously. Until recently, these techniques have been limited to small (1 to 10 wavelength) geometries because of computer run time and memory constraints. With the development of faster computers with more memory, the MM technique has increasing application to larger geometries. However, computer memory and run time can still be inadequate to solve many important antenna and scattering problems. Numerically efficient solutions require less computer memory and/or less computer run time. Therefore, any increase in the efficiency of a MM solution is of great practical interest.

The usual MM approach to modeling a thin wire of arbitrary shape is to specify a series of points, with piecewise linear segments between the points to approximate the wire. The current is represented by one or more basis functions, each with constant phase, over a piecewise linear segment. Typically, the size of the segments is set by how accurately the current or scattered field needs to be determined. For

convergence of the current, segment lengths of 0.05λ to 0.1λ are generally required.

A second factor that influences the segment size is the radius of curvature of the wire. Tightly curved wires require smaller segments to reproduce the wire shape accurately. When the radius of curvature is larger than a wavelength, the first case sets the segment size; when the radius of curvature is much less than a wavelength, the second case dictates the segment size (Figure 1).

All generally available MM codes based on the method of subdomains use the first approach. A natural question arises: Does dividing the wire into curved segments that conform to its shape, with arclengths restricted only by the maximum current variation rule, yield a solution that converges with fewer subsections? If the answer is yes, is the improvement in convergence worth the greater complexity and coding effort? To resolve this issue, the following approach is taken:



Figure 1. Left: Piecewise Linear Segments, Right: Curved Segments

1. Formulate the solution for linearly and circularly polarized plane wave incidence for a geometrically simple shape such as a loop.
2. Computer code the solution in FORTRAN.
3. Validate the solution using other MM solutions and measured data.
4. Study the convergence of the solution with respect to loop parameters and compare its performance to a method using linear subsections.
5. Study the effect of changes in program structure on its computational efficiency.

Chapter II discusses the derivation and the MM solution of the electric field integral equation for a thin wire loop. Chapter III discusses three MM FORTRAN programs used to determine the current on the loop. The entire domain solution (due to R. F. Harrington), which uses complex Fourier modes (HARLOOP) is considered to be the most accurate and therefore serves as a baseline for evaluating the other solutions. A second program that uses linear segments (LOOPSCAT) is compared to a third program that uses curved segments (CURVENEW). Chapter IV discusses the results obtained by the three methods and presents some guidelines in choosing an optimum solution method for a given antenna or scattering problem.

II. THE THIN WIRE INTEGRAL EQUATION

A. DERIVATION OF THE THIN-WIRE ELECTRIC FIELD INTEGRAL EQUATION

In this chapter, the integral equation for the current on a thin wire will be developed. Time-harmonic field quantities are assumed throughout. Phasor quantities are used with the $e^{j\omega t}$ dependency suppressed.

Referring to the thin wire geometry of Figure 2, the origin is point 0, the location of a source point is given by the vector \mathbf{r}' and an observation point by the vector \mathbf{r} . The wire radius, a , is considered constant over the length of the wire. The vector \mathbf{l} is everywhere parallel to the surface of the wire. Since the sum of the tangential components of the incident and scattered electric field must vanish at the surface of a perfect electric conductor, the boundary condition is,

$$\hat{\mathbf{n}} \times (\mathbf{E}^i + \mathbf{E}^s) = 0 \quad (2.1)$$

If the radius of the wire is small compared to the wavelength of the excitation, the surface current density, \mathbf{J}_s , can be considered constant around the circumference of the wire and directed along its axis. The excitation field can be either an incident wave or an impressed voltage. The scattered field is the field due to the current on the conductor induced by the excitation field.

The wave equation in terms of the vector potential \mathbf{A} is given by

$$\nabla^2 \mathbf{A} + \beta^2 \mathbf{A} = -\mu \mathbf{J}_s \quad (2.2)$$

where $\beta = 2\pi/\lambda$. The solution to equation (2.2) is

$$\mathbf{A} = \frac{\mu}{4\pi} \int_s \frac{\mathbf{J}_s e^{j\beta|\mathbf{r}-\mathbf{r}'|}}{|\mathbf{r}-\mathbf{r}'|} dS' = \mu \int_s \mathbf{J}_s g(\mathbf{r}, \mathbf{r}') dS' \quad (2.3)$$

where the integration is over the primed (source) coordinates. The Green's function, $g(\mathbf{r}, \mathbf{r}')$ is defined as,

$$g(\mathbf{r}, \mathbf{r}') = \frac{e^{-j\beta|\mathbf{r}-\mathbf{r}'|}}{4\pi|\mathbf{r}-\mathbf{r}'|} . \quad (2.4)$$

The expression for the scattered electric field in terms of \mathbf{A} is,

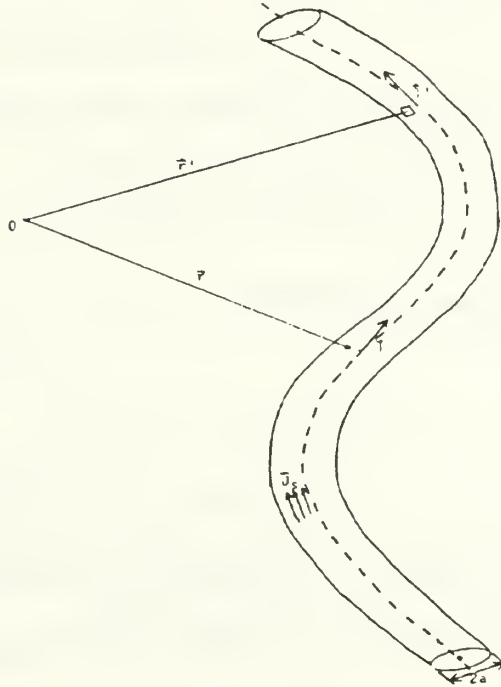


Figure 2. Thin Wire Geometry

$$\mathbf{E}^s = -j\omega\mathbf{A} - \frac{j}{\omega\mu\epsilon}\nabla(\nabla\cdot\mathbf{A}) . \quad (2.5)$$

Applying the boundary condition of equation (2.1),

$$\mathbf{E}_{\text{tan}}^i = -\mathbf{E}_{\text{tan}}^s = j\omega\mathbf{A} + \frac{j}{\omega\mu\epsilon}\nabla(\nabla\cdot\mathbf{A}) \quad \text{on } S . \quad (2.6)$$

Substitution of \mathbf{A} in equation (2.3) into equation (2.6) gives,

$$\mathbf{E}_{\text{tan}}^i = j\omega\mu\int_{s'}\mathbf{J}_s g(\mathbf{r},\mathbf{r}')dS' + \frac{j}{\omega\mu\epsilon}\nabla\nabla\cdot\left[\mu\int_{s'}\mathbf{J}_s g(\mathbf{r},\mathbf{r}')dS'\right] \quad \text{on } S. \quad (2.7)$$

Call the second term on the right side of equation (2.7) \mathbf{V} , and assume the medium to be homogeneous,

$$\mathbf{V} = \nabla\left[\nabla\cdot\mu\int_{s'}\mathbf{J}_s g(\mathbf{r},\mathbf{r}')dS'\right] = \nabla\left[\mu\int_{s'}\nabla\cdot(\mathbf{J}_s g(\mathbf{r},\mathbf{r}'))dS'\right]. \quad (2.8)$$

The vector identity for the divergence of a scalar u times a vector \mathbf{v} is,

$$\nabla\cdot(u\mathbf{v}) = \nabla u\cdot\mathbf{v} + u(\nabla\cdot\mathbf{v}) . \quad (2.9)$$

Applying this identity to equation (2.8) gives

$$\mathbf{V} = \nabla\left[\mu\int_{s'}(\nabla g(\mathbf{r},\mathbf{r}'))\cdot\mathbf{J}_s dS'\right]. \quad (2.10)$$

It can be shown that $\nabla g(\mathbf{r},\mathbf{r}') = -\nabla'g(\mathbf{r},\mathbf{r}')$ [Ref. 2]. Using this in equation (2.10)

and applying the identity of equation (2.9) again yields,

$$\begin{aligned}
V &= -\nabla \left[\mu \int_{s'} (\nabla' g(\mathbf{r}, \mathbf{r}') \cdot \mathbf{J}_s) dS' \right] \\
&= \nabla \left[\mu \int_{s'} g(\mathbf{r}, \mathbf{r}') (\nabla' \cdot \mathbf{J}_s) dS' - \mu \int_{s'} \nabla' \cdot (g(\mathbf{r}, \mathbf{r}') \mathbf{J}_s) dS' \right]
\end{aligned} \tag{2.11}$$

where ∇' is the del operator defined with respect to the primed coordinates. The second integral on the right side of equation (2.11) is equal to zero by the surface divergence theorem [Ref. 3]. Thus, V simplifies to

$$V = \mu \int_{s'} (\nabla' \cdot \mathbf{J}_s) \nabla g(\mathbf{r}, \mathbf{r}') dS' . \tag{2.12}$$

Substitution of equation (2.12) into equation (2.7) gives an integral equation for \mathbf{J}_s ,

$$\mathbf{E}_{\text{tan}}^i = j\omega\mu \int_{s'} \mathbf{J}_s g(\mathbf{r}, \mathbf{r}') ds' + \frac{j}{\omega\epsilon} \int_{s'} (\nabla' \cdot \mathbf{J}_s) \nabla g(\mathbf{r}, \mathbf{r}') dS' \tag{2.13}$$

which may be expressed more compactly as

$$\mathbf{E}_{\text{tan}}^i = \int_{s'} \left[j\omega\mu \mathbf{J}_s g(\mathbf{r}, \mathbf{r}') + \frac{j}{\omega\epsilon} (\nabla' \cdot \mathbf{J}_s) \nabla g(\mathbf{r}, \mathbf{r}') \right] dS' . \tag{2.14}$$

Equation (2.14) is a form of the Electric Field Integral Equation (EFIE). The unknown quantity to be solved for is \mathbf{J}_s .

B. SOLUTION OF THE EFIE USING MM

The method of moments (MM) technique can be used to solve for \mathbf{J}_s by expanding it into a series of basis functions, \mathbf{J}_i ,

$$J_s = \sum_i^N C_i J_i \quad (2.15)$$

where the C_i are complex constants to be determined. Substitution of equation (2.15) into 2.14 gives

$$E_{\text{tan}}^i = \sum_{i=1}^N C_i \int_{S_i} \left[j\omega\mu J_i g(r, r') + \frac{j}{\omega\epsilon} (\nabla' \cdot J_i) \nabla g(r, r') \right] dS' . \quad (2.16)$$

To generate the required N equations to solve for the N unknowns, define a suitable weighting function W_k , and take the inner product of W_k with both sides of equation 2.16. The inner product is defined such that it satisfies

$$\begin{aligned} \langle w, v \rangle &= \langle v, w \rangle \\ \langle \alpha f + \gamma v, w \rangle &= \alpha \langle f, w \rangle + \gamma \langle v, w \rangle \\ \langle v^*, v \rangle &> 0 \quad \text{if } v \neq 0 \\ \langle v^*, v \rangle &= 0 \quad \text{if } v = 0 \end{aligned} \quad (2.17)$$

[Ref. 4]. Choose the following inner product:

$$\langle w, v \rangle = \int_S w^* \cdot v dS \quad (2.18)$$

where $*$ signifies the complex conjugate. This leads to

$$\int_{S_k} W_k \cdot E_{\text{tan}}^i dS = \int_{S_k} j\omega\mu W_k \cdot \sum_{i=1}^N C_i \int_{S_i} \left[J_i g(r, r') + \frac{j}{\omega\epsilon} (\nabla' \cdot J_i) \nabla g(r, r') \right] dS' dS . \quad (2.19)$$

Interchanging the order of summation and integration and applying the surface divergence theorem yields for the right hand side of equation (2.19)

$$\sum_{i=1}^N C_i \int_{S_k} \int_{S_l} \left[j\omega\mu(\mathbf{W}_k \cdot \mathbf{J}_i)g(\mathbf{r}, \mathbf{r}') - \frac{j}{\omega\epsilon}(\nabla' \cdot \mathbf{J}_i)(\nabla \cdot \mathbf{W}_k)g(\mathbf{r}, \mathbf{r}') \right] dS' dS . \quad (2.20)$$

By making the following definitions,

$$\begin{aligned} V_k &= \int_{S_k} \mathbf{W}_k \cdot \mathbf{E}^i_{\text{tan}} dS \\ Z_{ik} &= \int_{S_k} \int_{S_l} \left[j\omega\mu(\mathbf{W}_k \cdot \mathbf{J}_i)g(\mathbf{r}, \mathbf{r}') - \frac{j}{\omega\epsilon}(\nabla' \cdot \mathbf{J}_i)(\nabla \cdot \mathbf{W}_k)g(\mathbf{r}, \mathbf{r}') \right] dS' dS \end{aligned} \quad (2.21)$$

2.20 and 2.21 can be written in matrix form,

$$[V] = [Z][C] \quad (2.22)$$

where $[V]$, $[Z]$ and $[C]$ are called the generalized voltage, impedance and current matrices, respectively. The unknown $[C]$ may be solved by an appropriate matrix inversion algorithm. Symbolically,

$$[C] = [Z]^{-1}[V] . \quad (2.23)$$

The generalized current matrix elements are the weighting coefficients in the summation of equation (2.15). The current \mathbf{J}_i is computed from equation (2.15) and the scattered field is calculated using this current in the radiation integrals. It should be noted that $[V]$, $[Z]$, and $[C]$ have units of volts, ohms, and amperes, but are not unique. In general, they depend on the choice of basis and weighting functions. However, the current will converge to the same numerical value as the number of basis functions are increased, provided the solutions are implemented correctly.

C. SPECIALIZATION OF THE EFIE TO A CIRCULAR LOOP USING CONFORMAL SUBSECTIONS.

The MM procedure will now be applied to a circular loop in the X-Y plane as illustrated in Figure 3. The loop is an ideal test geometry to study the characteristics of a MM solution using conformal subsections. It is a relatively simple geometry and other accurate solution methods are available to evaluate the performance of conformal subsections. The loop has a radius r_0 and is divided into N conformal segments. In this case a conformal segment is a circular arc. The arclength of the i^{th} segment is,

$$\Delta l_i = l_{i+1} - l_i = r_0 \Delta \phi_i . \quad (2.24)$$

The basis functions, J_i , of equation (2.15) are chosen to be overlapping triangles,

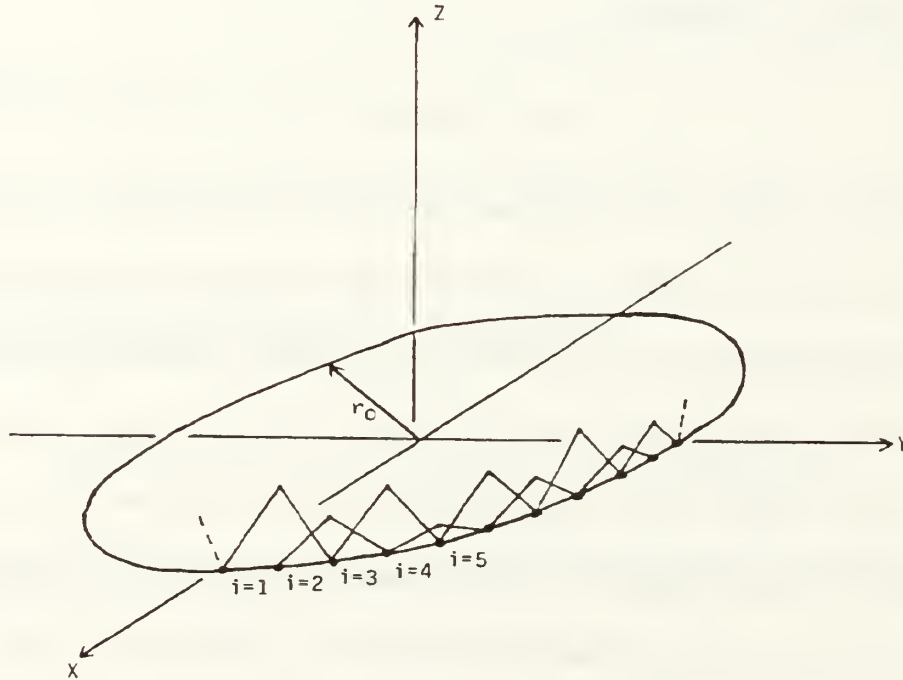


Figure 3. Thin Wire Loop Geometry

$$\mathbf{J}_i = T_i(l)\hat{\mathbf{l}} = \begin{cases} \frac{\hat{\mathbf{l}}(l-l_i)}{2\pi a \Delta l_i}; & l_i < l \leq l_{i+1} \\ \frac{\hat{\mathbf{l}}}{2\pi a} \left(1 - \frac{l-l_{i+1}}{\Delta l_{i+1}}\right); & l_{i+1} < l \leq l_{i+2} \\ 0; & \text{elsewhere} \end{cases} \quad (2.25)$$

Triangular basis functions are chosen because they are a more accurate representation of the current than a pulse basis function since the current is continuous everywhere along the wire, and they are relatively easy to deal with analytically. Balanis [Ref. 5] states that increasing the basis function complexity beyond triangles may not be warranted by the additional improvement in convergence rate. The triangular basis functions span two segments and overlap as shown in Figure 3. Therefore, the resultant current will be piecewise linear. The weighting (or testing) functions \mathbf{W}_k in equation (2.19) are chosen such that $\mathbf{W}_k = \mathbf{J}_k$ (Galerkin's method). Wang [Ref. 6] states that Galerkin's method provides numerical results which are more accurate than other testing methods under similar computational constraints. Substitution of the above weighting and basis functions in the expression for Z_{ik} in equation (2.21) yields

$$Z_{ik} = \int_{S_k} \int_{S_i} \left[j\omega \mu T_k(l) T_i(l') (\hat{\mathbf{l}} \cdot \hat{\mathbf{l}}') - \frac{j}{\omega \epsilon} T'_i(l') T'_k(l) \right] g(\mathbf{r}, \mathbf{r}') dS' dS \quad (2.26)$$

where $T'_i(l') = \frac{\partial T_i}{\partial l'}$, $T'_k(l) = \frac{\partial T_k}{\partial l}$.

Using the following relations,

$$\begin{aligned}
\hat{l} \cdot \hat{l}' &= \hat{\phi} \cdot \hat{\phi}' = \cos(\phi - \phi') \\
l &= r_0 \phi; \quad l' = r_0 \phi' \\
dS &= 2\pi a r_0 d\phi; \quad dS' = 2\pi a r_0 d\phi' \\
T_i(\phi) &= T_i(r_0 \phi) = T_i(l) \\
T'_i(l') &= \frac{1}{r_0} T'_i(\phi) \\
\eta &= \sqrt{\frac{\mu}{\epsilon}}; \quad \beta = \omega \sqrt{\mu \epsilon} \\
g(r, r') &= \frac{e^{-j\beta |R|}}{4\pi |R|}; \quad R = r - r'
\end{aligned} \tag{2.27}$$

equation (2.26) may be written as

$$Z_{ik} = \frac{r_0^2 j \beta \eta}{4\pi} \int_{\phi_k}^{\phi_{k-2}} \int_{\phi_i}^{\phi_{i-2}} \left[T_k(\phi) T_i(\phi') \cos(\phi - \phi') - \frac{1}{\beta^2 r_0^2} T'_i(\phi') T'_k(\phi) \right] \frac{e^{-j\beta |R|}}{|R|} d\phi d\phi' \tag{2.28}$$

From Figure 4 and the law of cosines, $|R|$ is given by,

$$|R|^2 = 2r_0^2 [1 - \cos(\phi - \phi')] + a^2 \tag{2.29}$$

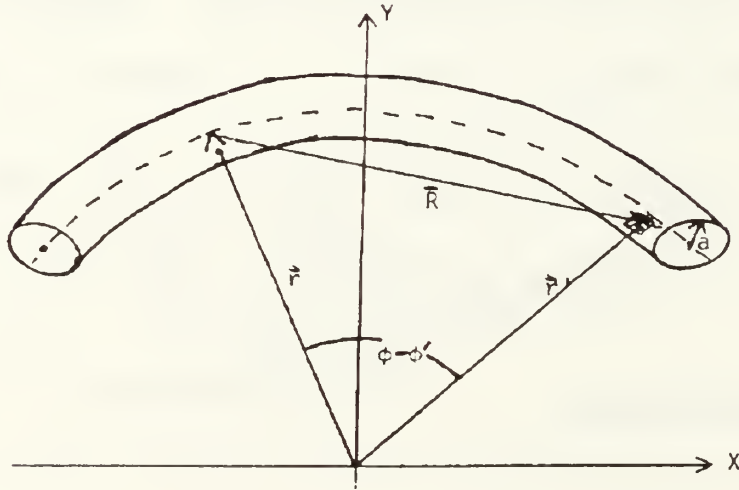


Figure 4. Geometry for Determining R.

and using the trigonometric identity,

$$\sin^2\left(\frac{\alpha}{2}\right) = \frac{1}{2}(1 - \cos\alpha) \quad (2.30)$$

results in,

$$|\mathbf{R}| = r_0 \sqrt{4 \sin^2\left(\frac{\phi - \phi'}{2}\right) + \frac{a^2}{r_0^2}} . \quad (2.31)$$

By choosing the test (unprimed) points at the center of the wire and the source points on the surface of the wire, the singularities along the diagonal of $[Z]$ at $\phi = \phi'$ where $\mathbf{r} = \mathbf{r}'$ are avoided. The technique used to calculate $|\mathbf{R}|$ is discussed further in Chapters III and IV.

The voltage elements, V_k , given in equation (2.21) become

$$V_k = r_0 \int_{r_0\phi_k}^{r_0\phi_{k+2}} \mathbf{T}_k(r_0\phi) \cdot \mathbf{E}^i d\phi . \quad (2.32)$$

The incident field, \mathbf{E}^i , for the purpose of this study, will be a plane wave. Figure 5 shows the direction of incidence of the plane wave in spherical polar angles $\theta = \Theta$ and $\phi = \Phi$ measured from the Z and X axes, respectively. \mathbf{E}^i can be θ or ϕ polarized. Referring to Figure 5, for θ polarization, the component of \mathbf{E}^i tangential to the loop, is

$$E_\theta^i \hat{\Theta} \cdot \hat{\Phi} = E_\theta^i \cos\Theta \sin(\Phi - \phi) . \quad (2.33)$$

Similarly, for ϕ polarization,

A similar expression for a ϕ directed incident field is

$$V_{\phi k} = r_0 E_{\phi}^i \int_{\phi_k}^{\phi_{k+2}} T_k(\phi) \cos(\Phi - \phi) e^{-j\beta r_0 \sin\Theta \cos(\Phi - \phi)} d\phi . \quad (2.37)$$

The computer coding of the solution for the thin wire loop using the equations developed above is described in the next chapter.

III. COMPUTER CODES FOR THE THIN WIRE LOOP

In this section, the FORTRAN program for a thin wire loop using curved segments is discussed. The results are presented and compared to similar solutions using straight subsections and Fourier modes.

A. DESCRIPTION OF THE CODES

Table 1 summarizes the organization of the three programs. Computer listings are given in Appendix B and equations from Chapter II will be referenced with a "2." preceding the equation number. The FORTRAN source code for the conformal subsections is named CURVENEW, and the codes for the straight subsections and the Fourier mode solution are named LOOPSCAT and HARLOOP respectively.

CURVENEW, LOOPSCAT and HARLOOP are functionally similar. Each calculates the loop geometry based on the segment size, loop radius, and wire radius, and each uses Gaussian quadrature for numerical integration. CURVENEW computes the impedance matrix, $[Z]$, in subroutine ZCURVED from the loop geometry of Figure 5 using equation (2.28). LOOPSCAT uses a similar formulation applied to straight segments in subroutine ZMATWW. HARLOOP computes $[Z]$ using the equations developed in reference [7]. CURVENEW computes the excitation vector, $[V]$, using equation (2.36) or (2.37) in subroutine CURVEW. LOOPSCAT uses a similar formulation for straight subsections in subroutine PLANEW. HARLOOP computes $[V]$

using the equations in reference [7] in subroutine PLANEW. In CURVENEW, all integrals are evaluated numerically and symmetry of the impedance elements is used to fill the $[Z]$ matrix and reduce the number of numerical calculations. Two formulations were investigated with LOOPSCAT: One using a delta function approximation to evaluate one of the double integrals in equation (2.21) and the other using Gaussian quadrature for both integrations. CURVENEW does all numerical integrations using Gaussian quadrature. Matrix symmetry is also used in LOOPSCAT to reduce the number

TABLE 1. FUNCTIONAL SUMMARY OF PROGRAMS

Program--> Function	CURVENEW (Curved Segments)	LOOPSCAT (Straight Segments)	HARLOOP (Fourier modes)
Read Input Parameters	Lines 25-38	Lines 15-27	Lines 19-30
Establish Loop Geometry	Lines 45-71	Lines 53-76	Lines 44-54
Calculate [Z]	Subroutine ZCURVED	Subroutine ZMATWW	Subroutine ZMATWW
Calculate [V]	Subroutine CURVEW	Subroutine PLANEW	Subroutine PLANEW
Solve System [C]=[Z] ⁻¹ [V]	Subroutines DECOMP and SOLVE	Subroutines DECOMP and SOLVE	Subroutines DECOMP and SOLVE
Calculate E^S	Lines 140-180	Lines 160-197	Lines 93-137

of calculations for [Z]. Computation of [C] is performed in subroutines DECOMP and SOLVE [Ref. 3], which solve the system of equations using Gaussian elimination. Subroutines DECOMP and SOLVE are common to all three programs. The subroutines ZCURVED and CURVIEW are discussed in more detail in the next section.

1. Loop Geometry

To generate the loop geometry an initial estimate of the desired circular arc length, Δl , is provided. This is used to estimate an angular increment, $\Delta\phi$,

$$\Delta\phi = \frac{\Delta l}{r_0} . \quad (3.1)$$

From this estimate, the number of generating points is calculated by,

$$N = \text{Int}\left(\frac{2\pi}{\Delta\phi}\right) + 1 \quad (3.2)$$

and $\Delta\phi$ is recalculated using,

$$\Delta\phi = \frac{2\pi}{N} \quad (3.3)$$

to ensure that $\Delta\phi$ is such that N segments fill exactly 2π radians. The loop points P_1 and P_2 are coincident with P_{N-1} and P_{N-2} so that the current is continuous around the loop.

2. Subroutines ZCURVED and ZMATWW

Subroutines ZCURVED and ZMATWW take advantage of the symmetry that exists on the loop. For all basis functions, the self impedance terms are equal. The

remainder of the matrix is filled with impedance values that repeat in a cyclic manner.

Mathematically,

$$\begin{aligned}
 Z_{11} &= Z_{22} = Z_{33} = \dots = Z_{NN} \\
 Z_{12} &= Z_{23} = Z_{34} = \dots = Z_{N-1\ N} \\
 Z_{13} &= Z_{24} = Z_{35} = \dots = Z_{N-2\ N} \\
 &\vdots \\
 &\vdots \\
 &\vdots \\
 Z_{1\ N-1} &= Z_{2\ N}
 \end{aligned} \tag{3.4}$$

The elements along any diagonal of $[Z]$ are equal and the lower off-diagonal elements are the mirror image of the upper diagonals. Thus $[Z]$ is a symmetrical Toeplitz matrix. Therefore, computation of the first row of $[Z]$ provides enough information to fill the entire matrix.

Because of the Green's function in the integrand for the impedance elements, the numerical treatment of the self term is very important. To optimize the convergence rate and accuracy of CURVENEW and LOOPSCAT, several different approaches are used to evaluate $|\mathbf{R}|$ near the singularity point where $\mathbf{r}=\mathbf{r}'$. In the first method, the observation point is chosen along the axis of the wire and the source point along the surface for all i,k , giving $|\mathbf{R}|=a$ at $\phi=\phi'$ (equation (2.31)). For the second method, both the observation point and the source point are chosen along the axis of the wire except on the segment $i=k$ where the value of ϕ at the midpoint is chosen on the axis of the wire, with \mathbf{r}' on the surface of the wire. Finally, both the observation point and the source point are chosen along the axis of the wire, except on the segment $i=k$, where \mathbf{r} is chosen along the axis, and \mathbf{r}' is chosen on the surface. Choosing the source point and

observation point as in the first case gives the most accurate results, but only slightly more accurate than the third case. Case two is accurate for small segment sizes but is inaccurate for larger segment sizes. Case three was selected for both CURVENEW and LOOPSCAT because it is only slightly less accurate than case one, and required fewer lines of code.

ZCURVED calculates the impedance elements of the first row of $[Z]$ by breaking the integral in equation (2.28) into four parts. For example, in the first row of $[Z]$, Z_{1i} , is calculated by summing contributions from the following four regions of integration (Figure 6):

1. A double integration along the positive slope of the T_1 over segment 1 and the positive slope of T_i over segment i .
2. A double integration along the negative slope of the T_1 over segment 2, and the positive slope of T_i over segment i .
3. A double integration along the positive slope of the T_1 over segment 1, and the negative slope of T_i over segment $i+1$.
4. A double integration along the negative slope of the T_1 over segment 2, and the negative slope of T_i over segment $i+1$.

The integrations are computed in a similar manner for the derivatives of the basis functions, T_1' and T_i' , over the same subsections. A similar procedure is used for the straight subsections in subroutine ZMATWW to calculate the impedance elements.

The numerical integrations are performed using Gaussian quadrature, with the number of points per subsection specified as an input parameter to program GAUSWGT, which computes the Legendre polynomial coefficients for a specified

number of integration points and writes them to file OUTGLEG. Gaussian quadrature was chosen because it requires fewer function evaluations than other methods for a given accuracy and does not require equal interval samples [Ref. 8]. CURVENEW, LOOPSCAT and HARLOOP read the coefficients from file OUTGLEG. The number of integration points per wavelength was varied to optimize convergence of LOOPSCAT and CURVENEW and is discussed in more detail in Chapter IV. The excitation vector [V] is calculated from equation (2.36) or (2.37) in subroutine CURVEW of CURVENEW.

3. Execution Time

Analysis of the nested DO loop structure of subroutine ZCURVED of program CURVENEW indicates that the total execution time of ZCURVED can be represented by,

$$T_{lc} = \alpha_c N_c N_{gc}^2 \quad (3.5)$$

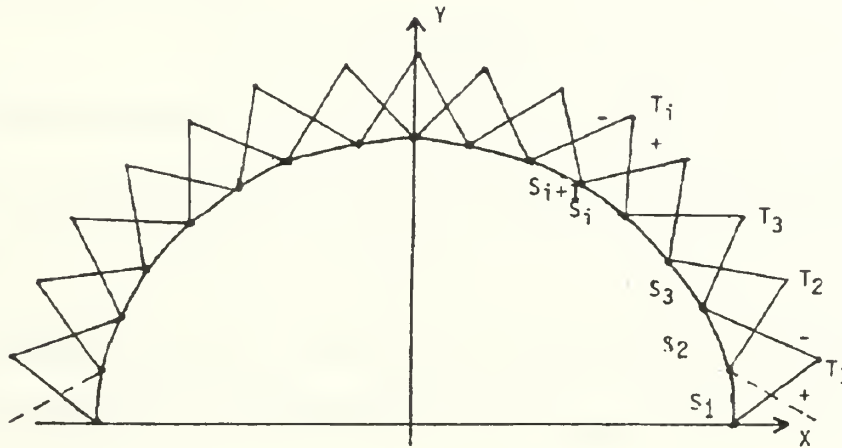


Figure 6. Loop Geometry for Impedance Integrations

where N_c is the number of curved segments (from equation (3.2)) N_{gc} is the number of Gaussian integration constants per curved segment and α_c is a constant. Execution time of the subroutines DECOMP and SOLVE, common to both CURVENEW and LOOPSCAT, can be represented similarly by [Ref. 9],

$$T_2 = \gamma N^3$$

where N is the number of segments. The excitation subroutine CURVEW execution time and field integrations are given by,

$$T_{3c} = \zeta_c N_c N_{gc} \quad (3.7)$$

where again γ and ξ_c are constants. Assuming that the execution time of the rest of the program is negligible, the total execution time of CURVENEW is

$$T_c = T_{1c} + T_2 + T_{3c} = \alpha_c N_c N_{gc}^2 + \gamma N_c^3 + \zeta_c N_c N_{gc} \quad (3.8)$$

A similar expression for LOOPSCAT uses the subscript l ,

$$T_l = \alpha_l N_l N_{gl}^2 + \gamma N_l^3 + \zeta_l N_l N_{gl} \quad (3.9)$$

Run times were recorded for various values of N_c , N_l and N_g using an IBM PC/AT with a math coprocessor and the coefficients for CURVENEW are found to be $\gamma=0.000156$, $\alpha_c=0.0230$, and $\xi_c=0.0222$. The coefficients for LOOPSCAT are $\alpha_l=0.0132$ and $\xi_l=0.0252$. For the moment, assume that the number of Gaussian integration points per wavelength, N_g , is held constant for both CURVENEW and LOOPSCAT. The number of integration points on a segment is,

$$N_{gc} = N_g \Delta l_c , \quad (3.10)$$

and the number of segments is,

$$N_c = \frac{2\pi r_0}{\Delta l_c} . \quad (3.11)$$

Similar expressions may be written for straight subsections. Combining equations (3.8), 3.10, and 3.11 gives

$$T_c = 4\pi^2 r_0^2 \frac{N_g^2 \alpha_c}{N_c} + \gamma N_c^3 + 2\pi r_0 \zeta_c N_g . \quad (3.12)$$

The ratio of T_c to T_l is given by,

$$T_c / T_l = \frac{4\pi^2 r_0^2 N_g^2 \alpha_c / N_c + \gamma N_c^3 + 2\pi r_0 \zeta_c N_g}{4\pi^2 r_0^2 N_g^2 \alpha_l / N_l + \gamma N_l^3 + 2\pi r_0 \zeta_l N_g} . \quad (3.13)$$

Equation (3.13) will be used in Chapter IV to compare the execution times of CURVENEW and LOOPSCAT.

IV. CALCULATED DATA FOR THE LOOP

The convergence of the MM solutions for both the current and electric field for circular loops of various dimensions are presented for both linear and circular polarizations. The convergence is shown to depend on the segment size and number of integration points, as well as excitation conditions (incidence direction and polarization). Representative plots are presented within the chapter, and additional plots are given in Appendix B.

A. CONVERGENCE OF HARLOOP

The Fourier mode solution, HARLOOP, was tested for convergence with respect to incidence angle, number of modes, and number of integration constants to establish a baseline for comparison to CURVENEW and LOOPSCAT. HARLOOP was chosen as a baseline because the sinusoidal basis functions match the physical behavior of the current on the loop, and thus the current series converges rapidly. This is illustrated in Figure 7 for a 0.5λ radius loop with a plane wave incident at an angle of 40 degrees.

Oscillation of the current as a function of ϕ becomes more rapid as Θ is increased, because the phase of the incident field over the loop varies as $\sin \Theta$ (equation (2.35)). For a θ polarized linear wave incident in the $\phi=0$ plane, the current is always zero at $\phi=0$ and 180 degrees, where \mathbf{E}^i is cross-polarized with respect to the axis of the wire. For θ polarized incident waves, maxima occur at $\phi=90$ and 270 degrees, where \mathbf{E}^i is

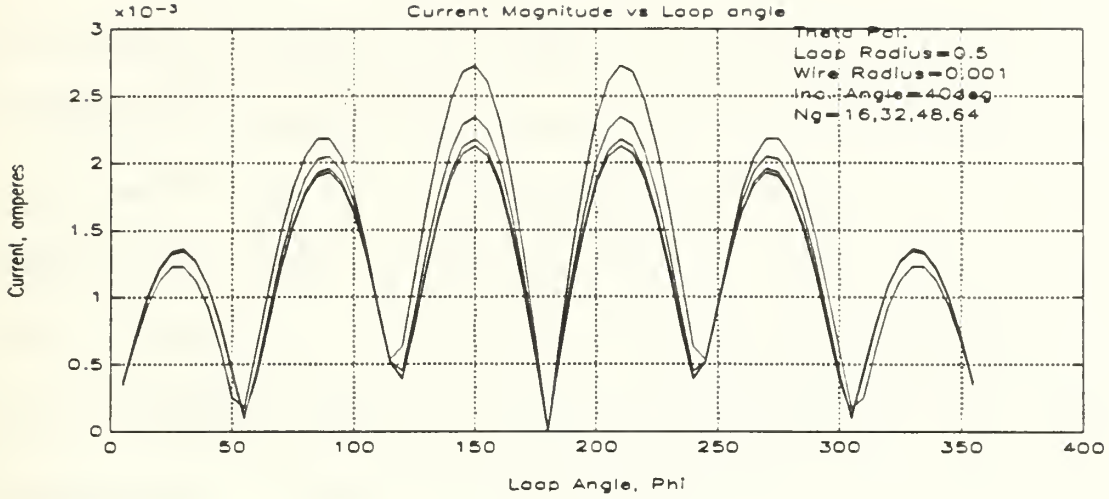


Figure 7. Convergence of the Current in the Complex Exponential Solution

parallel to the axis of the wire. The overall current amplitude decreases for Θ approaching 90 degrees, as expected, since the loop's projected area is small as viewed by the incident wave. For ϕ polarized incident waves, the minima occur at $\phi=90$ and 270 degrees and maxima at $\phi=0$ and 180 degrees. The currents do not vanish for Θ approaching 90 degrees because the loop is parallel to the ϕ polarized incident field. Circularly polarized incident waves give a constant magnitude current at normal incidence, and oscillations increase with Θ . For $\Theta=90$ degrees, the circular and ϕ polarization responses are identical.

HARLOOP is also found to be in agreement with measurements taken on the echo area of wire loops at normal incidence [Ref. 10]. The plot of Figure 8 gives the echo area (σ/λ^2) versus r_0 for varying wire radius using HARLOOP. Measured data is indicated by the '+' sign.

B. CONVERGENCE OF THE CURRENT EXPANSION

Having established HARLOOP as a baseline for comparison, convergence of the curved subsection program CURVENEW and the linear subsection program LOOPSCAT was evaluated. The plots of Figures 9 through 14 give the current on the loop as a function of loop angle, ϕ , for varying Θ , loop radius, and incident wave polarization. The number of integration points per wavelength, N_g , is held constant at 320 in LOOPSCAT and CURVENEW. This number was chosen empirically to give a converged current within five to ten percent RMS. The RMS error is defined relative to the Fourier mode solution. The HARLOOP current is plotted with the solid line, those of CURVENEW are plotted with the "+" sign, and those of LOOPSCAT with the "o". The wire radius is 0.001λ for these calculations.

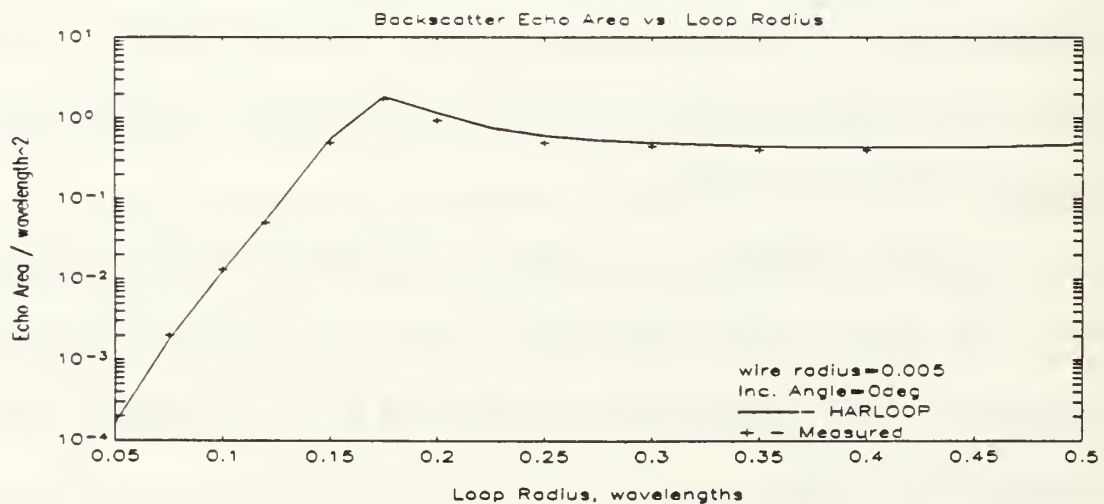


Figure 8. Backscatter Echo Area for a Loop with varying Radius at Normal Incidence

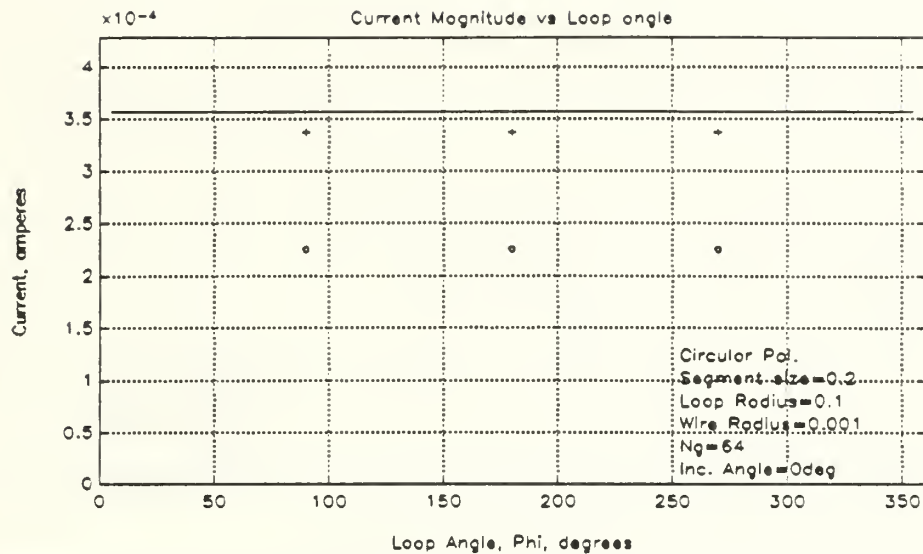


Figure 9. Magnitude of the Current on a 0.1λ Radius Loop, Normal Incidence, Circular Polarization (+ =curved; o =linear)

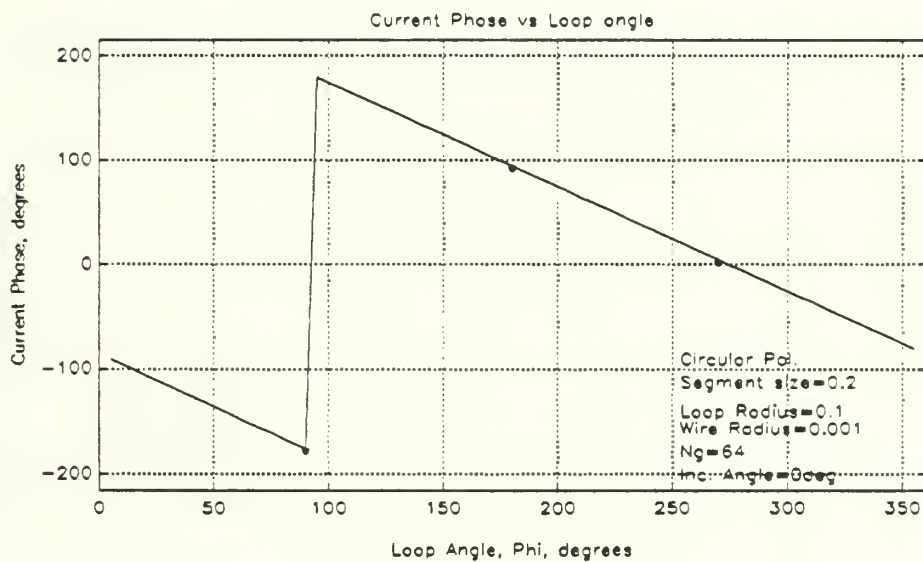


Figure 10. Phase of the Current on a 0.1λ Radius Loop, Normal Incidence, Circular Polarization (+ =curved; o =linear)

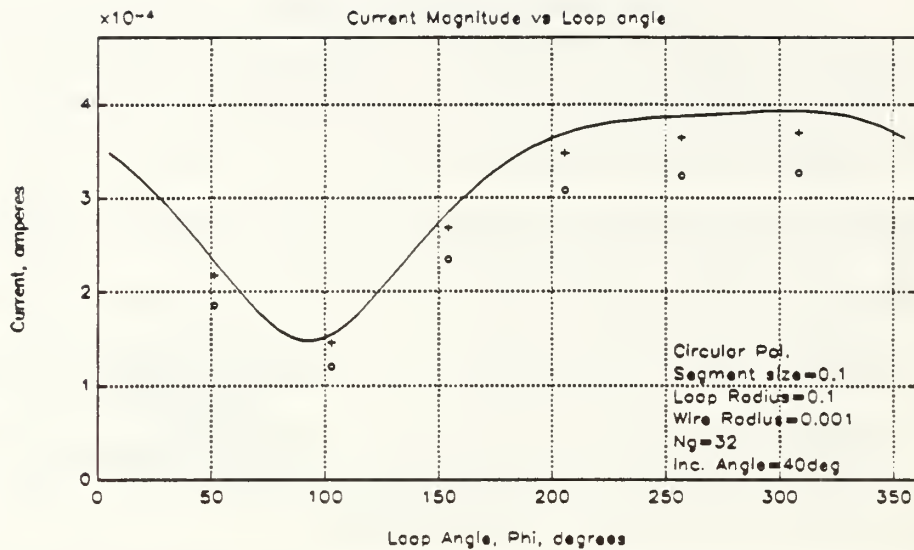


Figure 11. Magnitude of the Current on a 0.1λ Radius Loop, Incidence Angle=40 deg, Circular Polarization (+ =curved; o =linear)

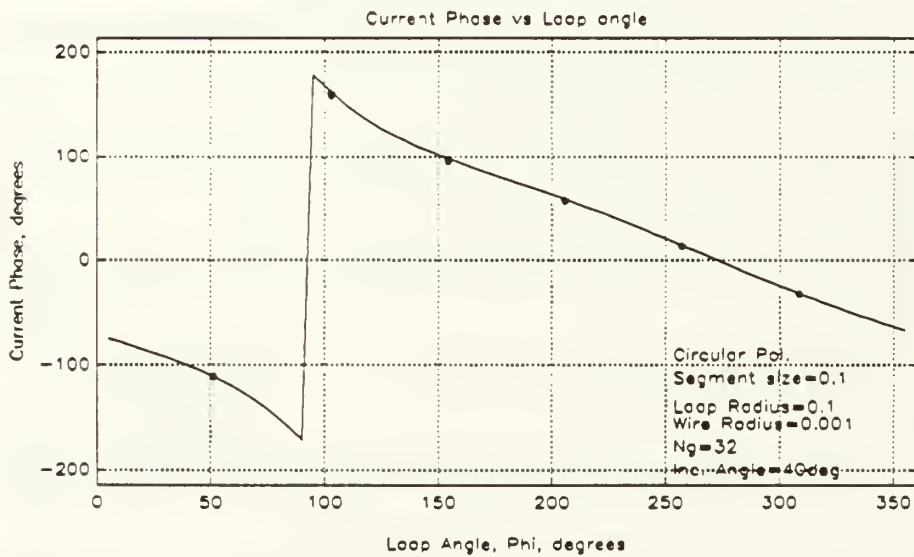


Figure 12. Phase of the Current on a 0.1λ Radius Loop, Incidence Angle=40 deg., Circular Polarization (+ =curved; o =linear)

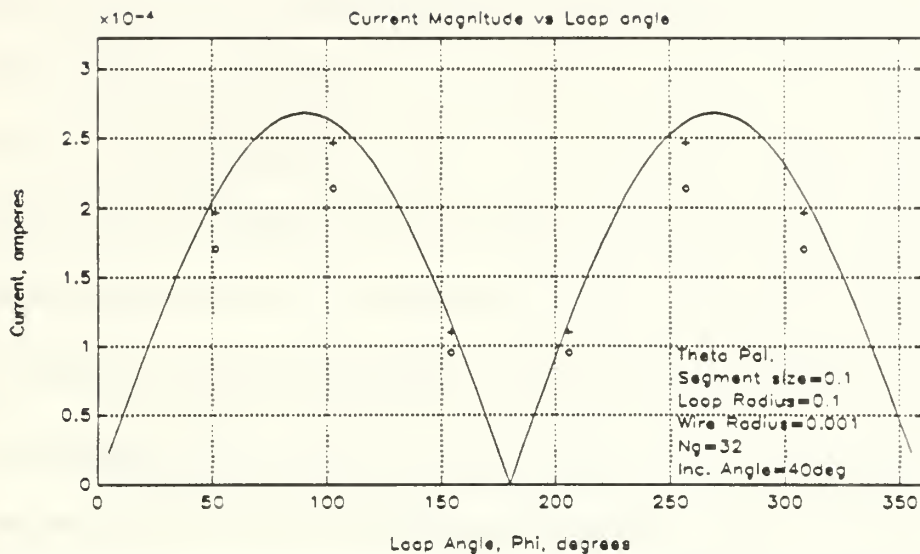


Figure 13. Magnitude of the Current on a 0.1λ Radius Loop, Incidence Angle=40 deg., Theta Polarization (+ =curved; o =linear)

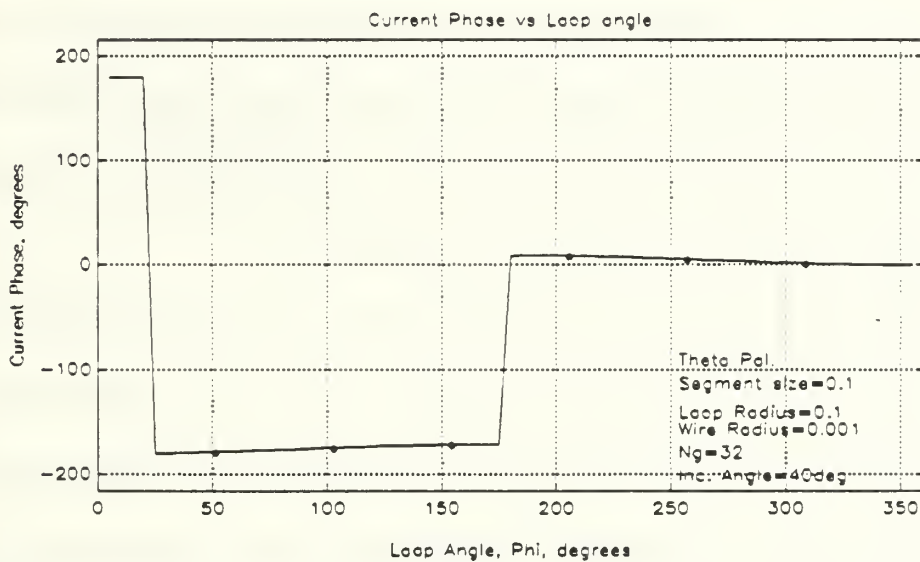


Figure 14. Phase of the Current on a 0.5λ Radius Loop, Incidence Angle=40 deg., Theta Polarization (+ =curved; o =linear)

Representative plots of the normalized root mean squared error are given in Figures 15 through 22. For set values of N and N_g , CURVENEW converges faster than LOOPSCAT in most cases. The error difference is most pronounced for loop circumferences on the order of a wavelength or less. From the plots, for $r_0=0.1 \lambda$, CURVENEW converges to less than 10 percent error for segment sizes ranging from 0.02 to 0.2 wavelengths ($N_c=32$ to $N_c=3$). LOOPSCAT converges to within 10 percent error for segment sizes less than approximately 0.06λ ($N_l > 10$) but gives errors of 30 to 40 percent for a segment size of 0.2 wavelengths. There is no improvement using curved subsections on larger loops for off-axis incidence waves, but the curved subsections give small improvements for large loops at normal incidence. This is expected in view of the behavior of the current on the loop. For linear polarization the current abruptly flips polarity from one side of the loop to the other. For circular polarization, the amplitude is constant, but the phase is linear. Both of these conditions can be represented accurately by a few triangles if the impedance and excitation integrals are evaluated precisely on the loop contour (see Figure 23).

Plots of the magnitude of the backscattered E^s versus incidence angle for varying segment size, N_g , and r_0 are given in Figures 24 and 25. As with the currents, the backscattered field converges more rapidly for CURVENEW than LOOPSCAT in most cases, with the greatest difference for small radius loops and angles near the maximum values of $|E^s|$.

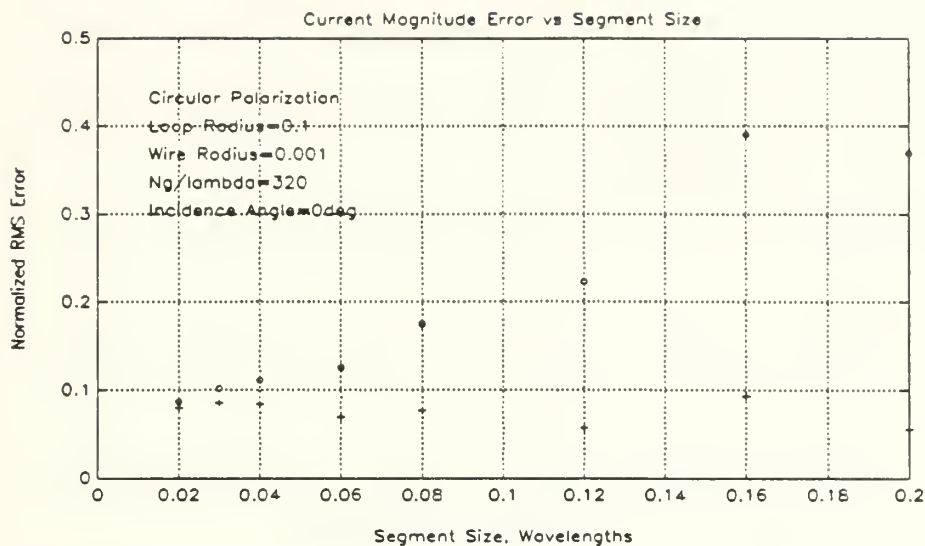


Figure 15. Error in the Current Magnitude for a 0.1λ Radius Loop, Normal Incidence, Circular Polarization (+ =curved; o =linear)

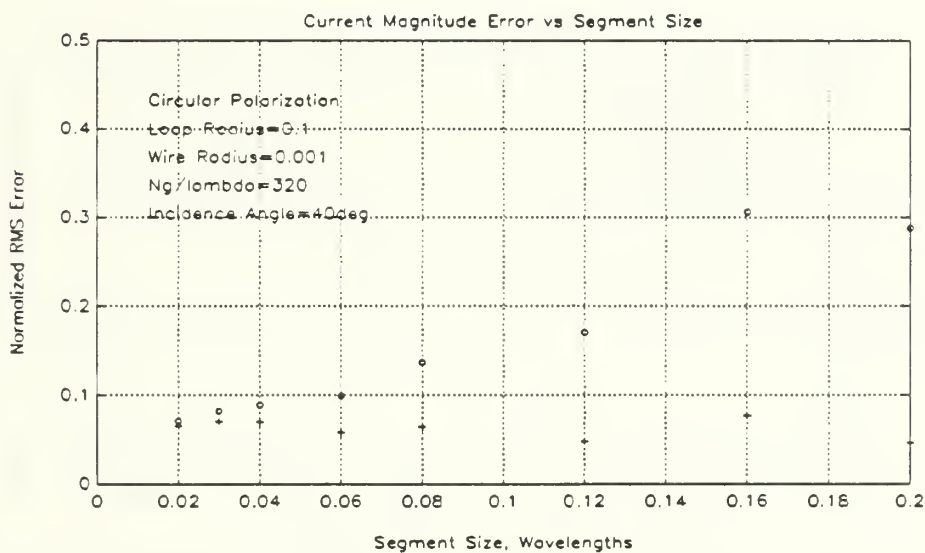


Figure 16. Error in the Current Magnitude for a 0.1λ Radius Loop, Incidence Angle=40 deg., Circular Polarization (+ =curved; o =linear)

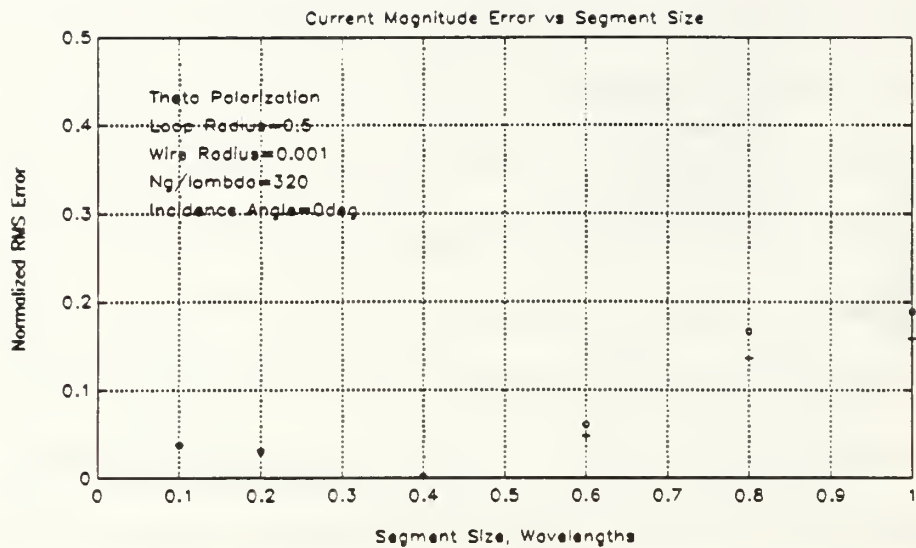


Figure 17. Error in the Current Magnitude for a 0.5λ Radius Loop, Normal Incidence, Circular Polarization (+ =curved; o =linear)

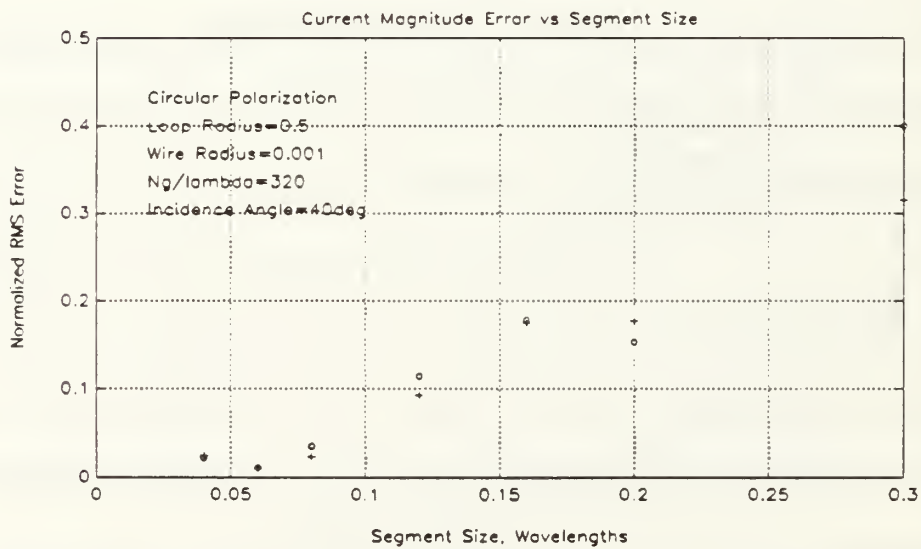


Figure 18. Error in the Current Magnitude for a 0.5λ Radius Loop, Incidence Angle=40 deg., Circular Polarization (+ =curved; o =linear)

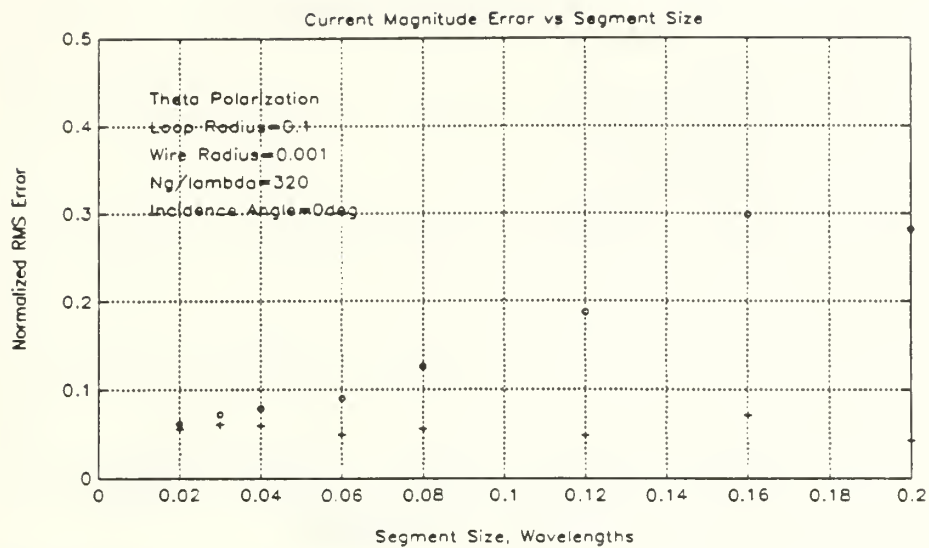


Figure 19. Error in the Current Magnitude for a 0.1λ Radius Loop, Normal Incidence, Theta Polarization (+ =curved; o =linear)

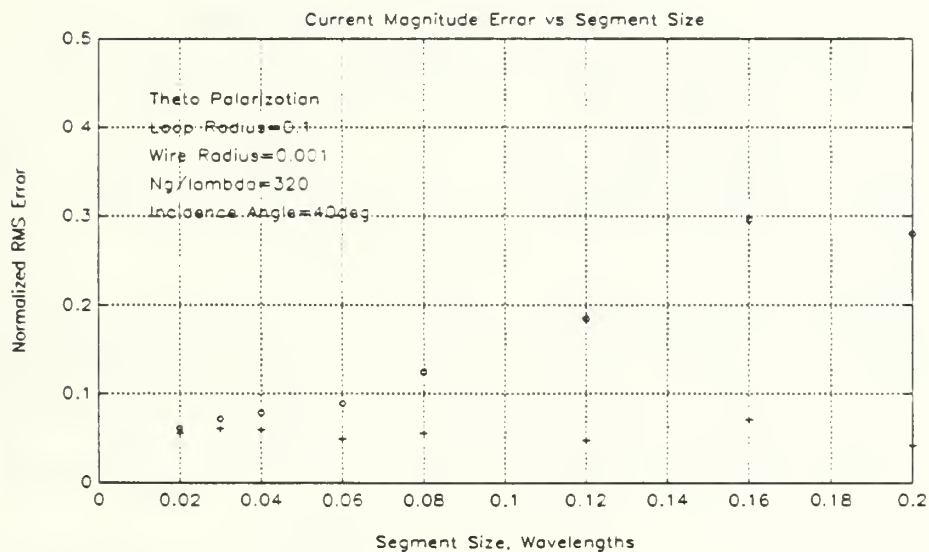


Figure 20. Error in the Current Magnitude for a 0.1λ Radius Loop, Angle of Incidence=40 deg., Theta Polarization (+ =curved; o =linear)

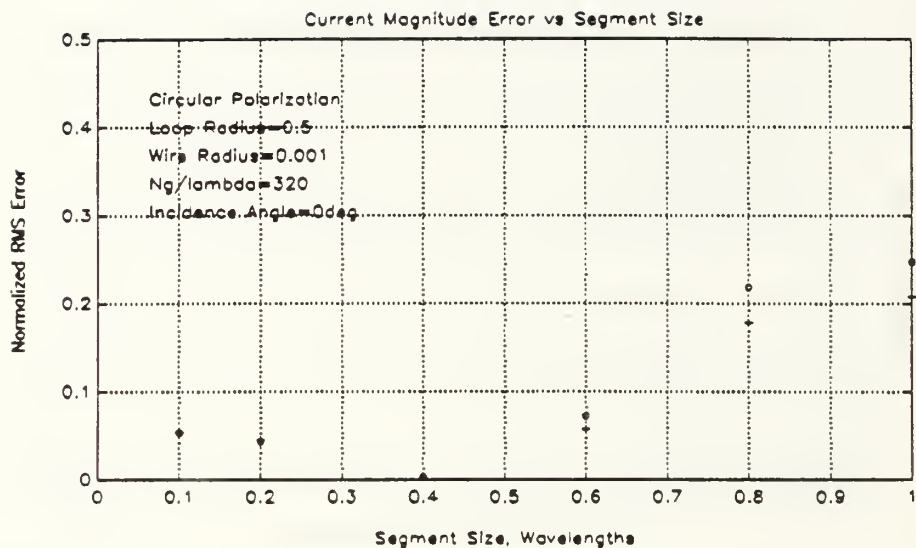


Figure 21. Error in the Current Magnitude for a 0.5λ Radius Loop, Normal Incidence, Theta Polarization (+ =curved; o =linear)

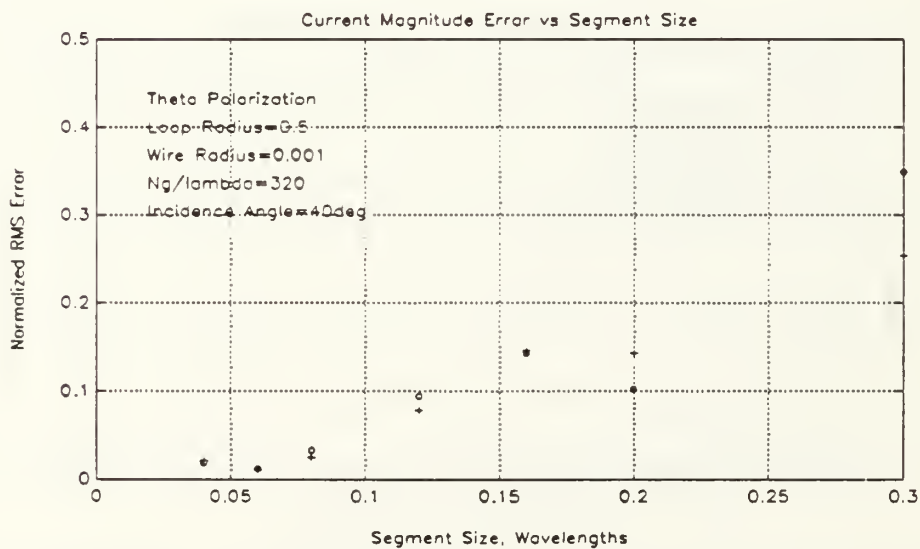


Figure 22. Error in the Current Magnitude for a 0.5λ Radius Loop, Incidence Angle=40 deg., Theta Polarization (+ =curved; o =linear)

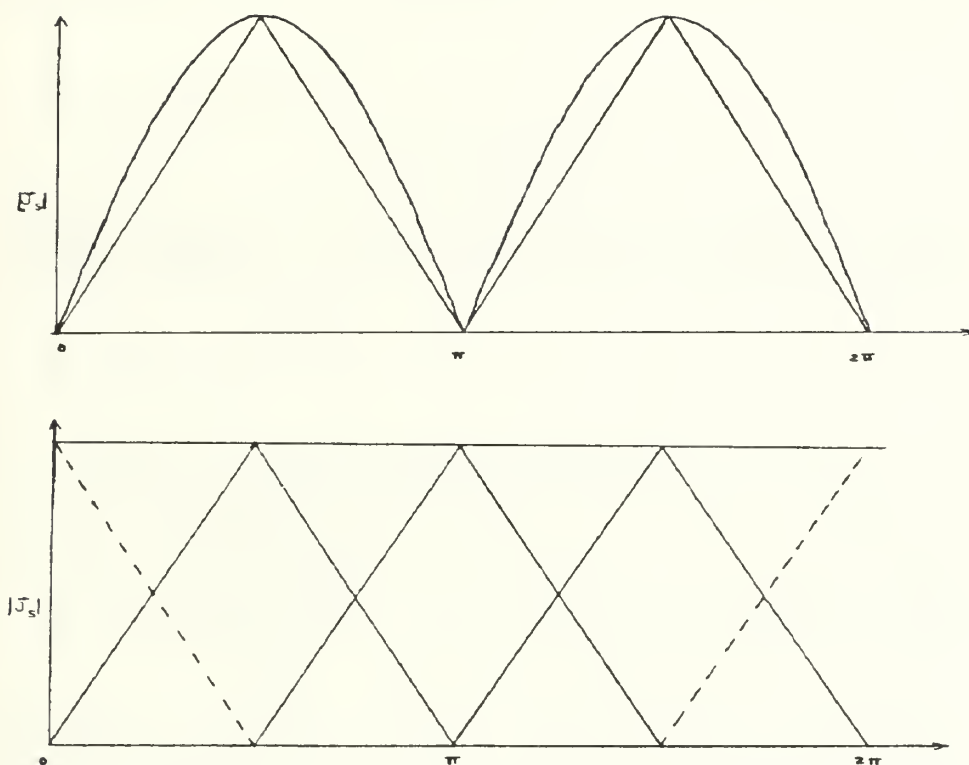


Figure 23. A Representation of a Sinusoidal Current with Two Basis Functions (top) and a Constant Current as a Superposition of Triangles (bottom)

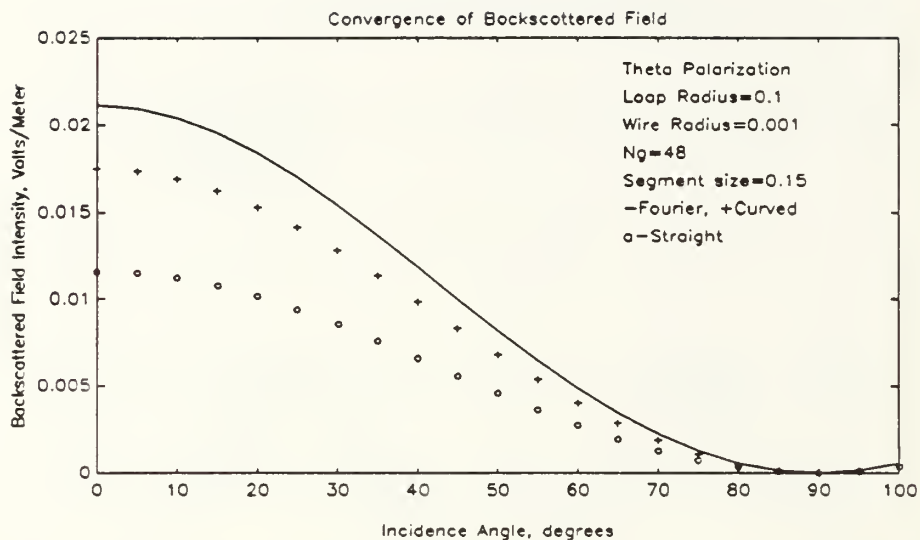


Figure 24. Backscattered Electric Field Intensity for varying Angles of Incidence, 0.1λ Radius Loop, Theta Polarization (+ =curved; o =linear)

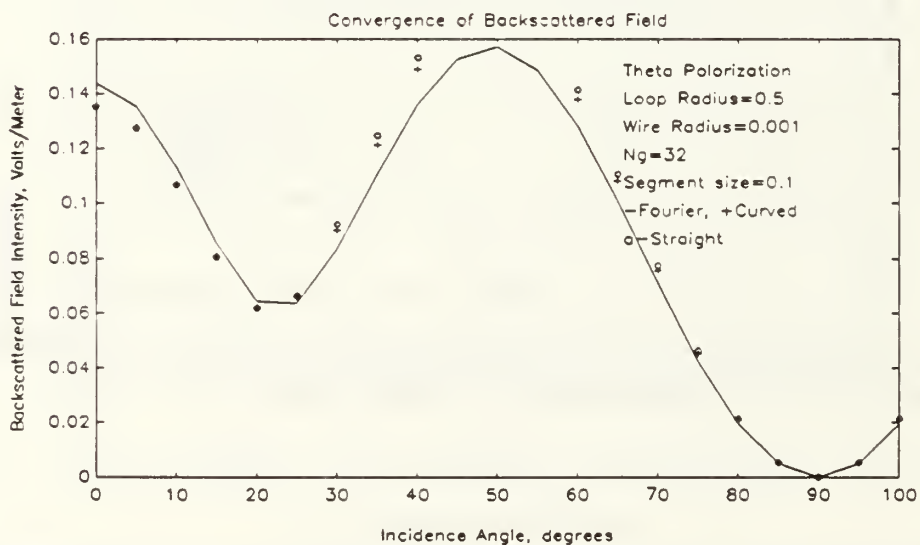


Figure 25. Backscattered Electric Field Intensity for varying Angles of Incidence, 0.5λ Radius Loop, Theta Polarization (+ =curved; o =linear)

C. COMPARISON OF EXECUTION TIME AND MEMORY REQUIREMENTS

The average run time for a given loop radius and convergence error is greater for CURVENEW than for LOOPSCAT due to the N_g^2 dependence in equation (3.13) and relative magnitudes of the coefficients α and γ . The plot of equation (3.13) in Figure 26 illustrates the ratio of run times of CURVENEW and HARLOOP versus N_l for $N_c=4$,

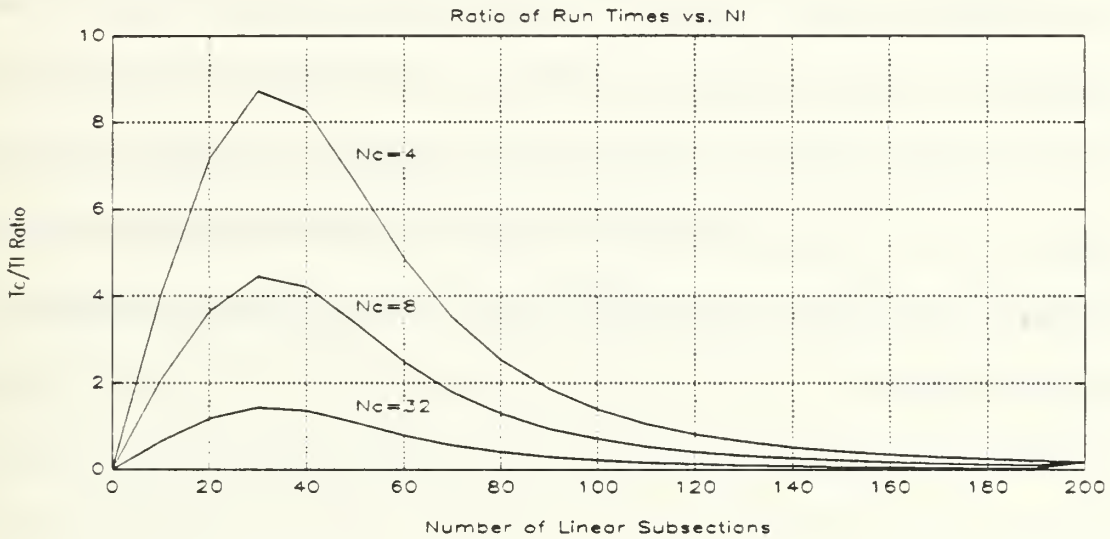


Figure 26. Ratio of Execution Times for CURVENEW and LOOPSCAT for Fixed Curved Segment Lengths and with Varying Linear Segment Lengths

8, 32 and $N_g=320$ for a 0.1λ loop. The run time of CURVENEW is less than HARLOOP for $T_c/T_l < 1$. From the plot, the "break even" points are approximately $N_l=115$, 90, 52 for $N_c=4$, 8, 32. For N_l less than these values, the integration subroutine ZCURVED is the determining factor in the run time; for N_l greater than these values, Gaussssian elimination subroutines DECOMP and SOLVE are the determining factors. The increased number of integrations per segment completely offset the time

savings of a reduced $[Z]$ matrix in CURVENEW. As mentioned in Chapter III, a delta function approximation for the outer integration of the impedance integral of equation (2.28) was investigated. This reduces the exponent of N_g to one in equation (3.13), but many more segments are required for a given accuracy. A more efficient integration scheme using a large number of integration points in the vicinity of $\phi = \phi'$ and fewer integrations elsewhere may reduce the execution time.

The savings in computer memory is significant for CURVENEW, since the number of matrix elements is on the order of N^2 . For a given accuracy for a 0.1λ loop, the ratio of the number of elements required for curved and linear subsections, $(N_c/N_l)^2$, is on the order of 0.1 to 0.2. This is a reduction of 80 to 90 percent. Although the memory requirements for the small loops considered here are not prohibitive for piecewise linear segments, greater memory savings will be realized for larger geometries comprised of many small features.

V. CONCLUSIONS

The use of conformal subdomain basis functions (curved subsections) to represent the current on a thin curved wire was investigated by solving the thin wire electric field integral equation using the method of moments. A solution using triangular basis functions was computer coded in FORTRAN and validated by comparing it to measured data and the results of two other method of moments solutions (LOOPSCAT and HARLOOP). The effect of varying loop radius, segment size, number of integration points and incident wave parameters on the accuracy and rate of convergence of the current expansion and backscattered field was investigated.

For small loops with circumferences on the order of a wavelength, the number of segments required to converge to a given accuracy with the curved segments was as small as 20 percent of the number of linear segments required to converge to the same accuracy (see Table 2). From computed data, it was determined that the greatest reduction in the number of unknowns for curved subsections occurs for geometries where the current amplitude variations over the surface are small and the phase variations are small or linear. As mentioned in Chapter I, the general rule of thumb for the length of one segment is 0.05λ to 0.1λ , which corresponds to a phase variation of 20 to 40 degrees. This restriction in phase variation is the driving factor when choosing the segment size for geometrical features having radii of curvature of approximately $\lambda/2$ or larger. A piecewise linear segment is small enough to represent the geometry accurately

in this case. The phase restriction applies to curved segments as well, but the curved segments conform exactly to the wire, and hence for small loops there is no sacrifice in geometrical accuracy by choosing segment sizes of approximately 0.05λ or 20 electrical degrees. As the loop becomes larger, or the wavelength becomes smaller, the curved and straight subsection solutions become equivalent. The greatest advantage in using curved subsections to reduce the number of segments is for electrically small structures where small linear segments are required simply to reproduce the wire shape.

Although the number of segments was greatly reduced using conformal subsections, the execution time was increased due to the increased number of integration points per segment required for acceptable accuracy. To reduce the integration time, it is suggested that the number of integration points per wavelength be varied from a large number when evaluating the self term, to fewer points away from the self term. For certain geometries, symmetry could also be used to reduce the integration time.

To avoid singularities, the MM testing procedure was performed along the axis of the wire and the current constrained to the surface of the wire. A delta function approximation in the impedance integrations was found to reduce the time required to compute the impedance matrix, but as expected, required more segments for a given convergence accuracy.

A disadvantage in the formulation of CURVENEW is that an analytic expression for the curve is needed to perform the integrations for the impedance matrix and the excitation vector. The expressions will change each time a new curve is analyzed, and consequently, considerable effort will be required to modify the code every time a change

is made in the geometry. Programs that use linear segments are more flexible because they require only the coordinates of the points along the wire axis to generate the integration points.

The next logical step in testing the effectiveness of conformal subdomain basis functions is to formulate the solution for an equiangular spiral wire. The equiangular spiral is used in broadband antennas and has a simple mathematical form. For geometrical accuracy, the segment size in the piecewise linear formulation will be much smaller than 20 electrical degrees near the center of the spiral. Equal length conformal

TABLE 2. COMPARISON OF CURVENEW AND LOOPSCAT

	Curved Subsections		Linear Subsections	
	0.1 λ	0.5 λ	0.1 λ	0.5 λ
Loop Radius				
Number of Segments for 10% RMS Error, Normal Incidence	3	5	21	5
Number of Segments for 10% RMS Error, Off Axis Incidence	3	26	10	26
Execution Time*, 10% RMS Error, Normal Incidence	314 s	4669 s	32 s	2691 s
Execution Time*, 10% RMS Error, Off Axis Incidence	314 s	918 s	59 s	540 s
[Z] Matrix Size, 10 % RMS Error, Normal Incidence	9	25	441	25
[Z] Matrix Size, 10 % RMS Error, Off Axis Incidence	9	676	100	676

* Execution time measured with an IBM PC/AT.

segments may be used along the spiral arms, and it is anticipated that the number of required segments will be substantially reduced.

REFERENCES

1. Wang, J. J. H., *Generalized Moment Methods in Electromagnetics*, pp. 2-3, John Wiley & Sons, Inc., 1991.
2. Balanis, C. A., *Advanced Engineering Electromagnetics*, p 708, John Wiley & Sons, Inc., 1989.
3. Mautz, J. R. and Harrington, R. F., "Radiation and Scattering From Bodies of Revolution," *Applied Science*, v. 20, p.405, June 1969.
4. Balanis, C. A., *Advanced Engineering Electromagnetics*, p. 689, John Wiley & Sons, Inc., 1989.
5. Balanis, C. A., *Advanced Engineering Electromagnetics*, p. 684, John Wiley & Sons, 1989.
6. Wang, J. J. H., *Generalized Moment Methods in Electromagnetics*, pp. 40-48, John Wiley & Sons, 1991.
7. Harrington, R. F., *Field Computations by Moment Methods*, Macmillan, 1968.
8. Gerald, C. F., and Wheatly, P. O., *Applied Numerical Analysis*, pp. 299-303, Addison-Wesley Publishing Company, 1989.
9. Hornbeck, Robert W., *Numerical Methods*, pp. 91-95, Quantum Publishers Inc., 1975.
10. Richmond, J. H., "A Wire Grid Model for Scattering by Conducting Bodies", *IEEE Transactions on Antennas and Propagation*, v. AP-14, no. 6, pp 782-786.

APPENDIX A

COMPUTER CODES

```

1
2 C MAIN PROGRAM *CURVNEW.FOR*
3 C **** MORE NUMERICALLY EFFICIENT THAN CURVSUB.F ****
4 C PLANE WAVE SCATTERING FROM A CIRCULAR LOOP IN THE Z PLANE.
5 C METHOD OF MOMENTS WITH CURVED SUBSECTIONS
6 C **** RADAR CROSS SECTION CALCULATION ****
7 C
8     COMPLEX Z(25000),B(500),C(500),BP(500),CP(500),EX,EC
9     COMPLEX ET,EP,UC,RW(500),U0
10    DIMENSION ECP(400),IPS(500),ANG(400),EDP(400)
11    DIMENSION XG(128),AG(128),PH(500),DEL(500),PA(500)
12    DIMENSION ECV(500),EXV(500),PHC(500),PHX(500)
13    DATA PI/3.14159265/
14    DATA IPRINT/1/,ITEST/1/
15    RAD=PI/180.
16    ECX=0.
17    BK=2.*PI
18    ETA=377.
19    U0=(0.,0.)
20    UC=(0.,-1.)*ETA*BK/4./PI
21    CONST=16.*PI**3
22    PHIRD=0.
23 C
24 C READ INPUT AND PROGRAM CONTROL PARAMETERS
25 C
26     OPEN(1,FILE='PARAMLIST.DAT')
27     READ(1,*)ANGLE,DT,START,STOP,AW,R0,SEG,HARR,GCONST,XLOC,XIPOL
28     LOC=INT(XLOC)
29     IPOL=INT(XIPOL)
30     CLOSE(1)
31 C
32 C READ GAUSSIAN CONSTANTS
33 C
34     OPEN(2,FILE='OUTGLEG')
35     READ(2,*) NG
36     DO 1 I=1,NG
37         READ(2,*) XG(I),AG(I)
38 1    CONTINUE
39 C
40 C OUTCURV IS THE FILE THAT THE SCATTERED FIELD DATA IS WRITTEN TO
41 C ICURVOUT IS THE FILE THAT THE LOOP CURRENT DATA IS WRITTEN TO
42 C

```



```

43         OPEN(8,FILE='OURCURV.DAT')
44         OPEN(7,FILE='ICURVOUT.DAT')
45     C
46     C GENERATE THE LOOP POINTS
47     C
48     C
49     C CHOOSE THE NUMBER OF POINTS BASED ON THE VALUE OF SEG
50     C
51     C   AW=AW*R0
52         DPHI=SEG/R0
53         NP=INT(2.*PI/DPHI)+1
54         DPHI=360./FLOAT(NP)
55         PH(1)=0.
56         DO 10 I=2,NP+1
57             PH(I)=FLOAT(I-1)*DPHI*RAD
58             DEL(I-1)=(PH(I)-PH(I-1))
59             PA(I-1)=(PH(I)+PH(I-1))/2.
60     10 CONTINUE
61         NP=NP+2
62     C
63     C OVERLAP THE ENDS SO THAT CURRENT WILL BE CONTINUOUS ON THE LOOP
64     C
65         PH(NP)=BK+PH(2)
66         DEL(NP-1)=DEL(1)
67         PA(NP-1)=BK+PA(1)
68         MT=NP-2
69         DO 52 I=1,NP
70             XHB=R0*COS(PH(I))
71             YHB=R0*SIN(PH(I))
72     52 CONTINUE
73         WRITE(6,*) 'GEOMETRY DEFINED'
74         IF(ITEST.EQ.0) GO TO 98
75     C
76     C DEFINE DIMENSIONS OF THE IMPEDANCE MATRIX BLOCKS
77     C
78         WRITE(6,*) 'NP,MT=',NP,MT
79     C
80     C COMPUTE IMPEDANCE MATRIX ELEMENTS
81     C
82         CALL ZCURVED(NP,R0,PH,DEL,PA,NG,XG,AG,AW,Z,ZMAX)
83         DO 11 I=1,MT
84             CZ=CABS(Z(I))
85             AZ=ATAN2(AIMAG(Z(I)),REAL(Z(I))+1.E-8)/RAD
86     11 CONTINUE
87         WRITE(6,*) 'WIRE IMPEDANCE COMPUTED'
88     C
89     C PERFORM LU DECOMPOSITION
90     C
91         CALL DECOMP(MT,IPS,Z)
92         WRITE(6,*) 'Z DECOMPOSED'

```

```

93      C
94      C BEGIN FIELD CALCULATIONS
95      C
96      98  PHR0=PHIRD*RAD
97          IT=INT((STOP-START)/DT)+1
98          WRITE(7,*) IT,MT,0,0
99          DO 500 I=1,IT
100             THETA=FLOAT(I-1)*DT+START
101             THR=THETA*RAD
102             PHR=PHR0
103             IF(THETA.LT.180.) GO TO 99
104             THR=(360.-THETA)*RAD
105             PHR=PHR0+PI
106      99  CONTINUE
107          ET=U0
108          EP=U0
109      C
110      C COMPUTE THE EXCITATION VECTOR
111      C
112          CALL CURVEW(NP,R0,PH,DEL,PA,NG,XG,AG,THR,PHR,RW)
113          IF (LOC .EQ. 0) THEN
114      C
115      C CIRCULAR POLARIZATION IF LOC=1 ELSE LINEAR
116      C
117          IF(IPOL.EQ.1) THEN
118      C
119      C THETA POLARIZED INCIDENT WAVE (IPOL=1)
120      C
121          DO 101 L=1,MT
122      101      B(L)=RW(L)
123          ELSE
124      C
125      C PHI POLARIZED INCIDENT WAVE (IPOL=2)
126      C
127          ENDIF
128          IF(ITEST.EQ.0) GO TO 9998
129      C
130      C PERFORM GAUSSIAN ELIMINATION TO DETERMINE [C]
131      C
132          CALL SOLVE(MT,IPS,Z,B,C)
133          DO 210 L=1,MT
134              WRITE(7,*) L,L*DPHI,CABS(C(L)),ATAN2(REAL(C(L)),
135      *      AIMAG(C(L)))/RAD
136              ET=ET+RW(L)*C(L)
137      210      EP=EP+RW(L+MT)*C(L)
138          ELSE
139      C
140      C THETA POLARIZED INCIDENT WAVE
141      C
142          DO 221 L=1,MT

```

```

143      221      B(L)=RW(L)
144      C
145      C PHI POLARIZED INCIDENT WAVE
146      C PHASE SHIFT FOR CP IS PI/2.
147      C
148          DO 222 L=1,MT
149      222      BP(L)=RW(L+MT)*CEXP(CMPLX(0.,PI/2.))
150          CALL SOLVE(MT,IPS,Z,B,C)
151          CALL SOLVE(MT,IPS,Z,BP,CP)
152          DO 220 L=1,MT
153              WRITE(7,*) L,L*DPHI,CABS(C(L)+CP(L)),ATAN2(REAL(C(L)+CP(L)),
154          *      AIMAG(C(L)+CP(L)))/RAD
155              ET=ET+RW(L)*C(L)+RW(L)*CP(L)
156      220      EP=EP+RW(L+MT)*C(L)+RW(L+MT)*CP(L)
157          ENDIF
158          EC=ET*UC
159          EX=EP*UC
160          ANG(I)=THETA
161          ECV(I)=CABS(EC)
162          EXV(I)=CABS(EX)
163          ECR=REAL(EC)
164          ECI=AIMAG(EC)
165          EXR=REAL(EX)
166          EXI=AIMAG(EX)
167          PHC(I)=ATAN2(ECI,ECR+1.E-20)/RAD
168          PHX(I)=ATAN2(EXI,EXR+1.E-20)/RAD
169          ECX=AMAX1(ECX,ECV(I),EXV(I))
170      500 CONTINUE
171          WRITE(6,*) 'EMAX=' ,ECX
172          DO 600 I=1,IT
173              ECV(I)=AMAX1(ECV(I),1.E-10)
174              EXV(I)=AMAX1(EXV(I),1.E-10)
175              ECP(I)=(ECV(I)/ECX)**2
176              EDP(I)=(EXV(I)/ECX)**2
177              ECP(I)=AMAX1(ECP(I),.00001)
178              EDP(I)=AMAX1(EDP(I),.00001)
179              ECP(I)=10.*ALOG10(ECP(I))
180              EDP(I)=10.*ALOG10(EDP(I))
181      600 CONTINUE
182          SIGMA=(ECX**2)*CABS(UC)/(2.*BK)
183          SIGDB=10.*ALOG10(SIGMA)
184          WRITE(6,*) 'BACKSCATTER CROSS-SECTION, IN DB=' ,SIGMA,SIGDB
185      208      FORMAT(/,5X,'SIGMA/WAVL SQ=' ,E15.4,
186          * /,5X,'      IN DB=' ,F8.4)
187          DO 9000 L=1,IT
188              WRITE(8,*) ANG(L),ECV(L)
189      9000 CONTINUE
190      9998 STOP
191          END
192          SUBROUTINE SOLVE(N,IPS,UL,B,X)

```

```

193 C
194 C
195 C SUBROUTINE TO SOLVE SYSTEM OF EQUATIONS WITH COMPLEX
196 COEFFICIENTS.
197 C CALL 'DECOMP' FIRST. (FROM MAUTZ AND HARRINGTON)
198 C
199     COMPLEX UL(50000),B(200),X(200),SUM
200     DIMENSION IPS(500)
201     NP1=N+1
202     IP=IPS(1)
203     X(1)=B(IP)
204     DO 2 I=2,N
205     IP=IPS(I)
206     IPB=IP
207     IM1=I-1
208     SUM=(0.,0.)
209     DO 1 J=1,IM1
210     SUM=SUM+UL(IP)*X(J)
211     1 IP=IP+N
212     2 X(I)=B(IPB)-SUM
213     K2=N*(N-1)
214     IP=IPS(N)+K2
215     X(N)=X(N)/UL(IP)
216     DO 4 IBACK=2,N
217     I=NP1-IBACK
218     K2=K2-N
219     IPI=IPS(I)+K2
220     IP1=I+1
221     SUM=(0.,0.)
222     IP=IPI
223     DO 3 J=IP1,N
224     IP=IP+N
225     3 SUM=SUM+UL(IP)*X(J)
226     4 X(I)=(X(I)-SUM)/UL(IPI)
227     RETURN
228     END
229     SUBROUTINE DECOMP(N,IPS,UL)
230 C
231 C SUBROUTINE TO DECOMPOSE SYSTEM OF EQUATIONS.
232 C FROM MAUTZ AND HARRINGTON.
233 C
234     COMPLEX UL(50000),PIVOT,EM
235     DIMENSION SCL(200),IPS(200)
236     DO 5 I=1,N
237     IPS(I)=I
238     RN=0.
239     J1=I
240     DO 2 J=1,N
241     ULM=ABS(REAL(UL(J1)))+ABS(AIMAG(UL(J1)))
242     J1=J1+N

```

```

243         IF(RN-ULM) 1,2,2
244     1 RN=ULM
245     2 CONTINUE
246         SCL(I)=1./RN
247     5 CONTINUE
248         NM1=N-1
249         K2=0
250         DO 17 K=1,NM1
251             BIG=0.
252             DO 11 I=K,N
253                 IP=IPS(I)
254                 IPK=IP+K2
255                 SIZE=(ABS(REAL(UL(IPK))))+ABS(AIMAG(UL(IPK))))*SCL(IP)
256                 IF(SIZE-BIG) 11,11,10
257     10 BIG=SIZE
258         IPV=I
259     11 CONTINUE
260         IF(IPV-K) 14,15,14
261     14 J=IPS(K)
262         IPS(K)=IPS(IPV)
263         IPS(IPV)=J
264     15 KPP=IPS(K)+K2
265         PIVOT=UL(KPP)
266         KP1=K+1
267         DO 16 I=KP1,N
268             KP=KPP
269             IP=IPS(I)+K2
270             EM=-UL(IP)/PIVOT
271     18 UL(IP)=-EM
272             DO 16 J=KP1,N
273                 IP=IP+N
274                 KP=KP+N
275                 UL(IP)=UL(IP)+EM*UL(KP)
276     16 CONTINUE
277         K2=K2+N
278     17 CONTINUE
279         RETURN
280         END
281     SUBROUTINE ZCURVED(NP,R0,PH,DEL,PA,NG,XG,AG,A,Z,ZMAX)
282 C
283 C IMPEDANCE ELEMENTS FOR CURVED BASIS FUNCTIONS.
284 C SPECIFICALLY DERIVED FOR A CIRCULAR LOOP -- NOT A GENERAL CASE.
285 C
286     COMPLEX CEXP,Z(50000),CON,CMPLX,SUMA,SUMB
287     COMPLEX U0,SUM1,SUM2,SUM3,SUM4,EXP,ZT(500)
288     DIMENSION DEL(500),PH(500),XG(128),AG(128),PA(500)
289 C   OPEN(2,FILE='ZCURV.DAT')
290     ETA=377.
291     ZMAX=0.
292     PI=3.14159

```

```

293         BK=2.*PI
294         BK2=BK**2
295         U0=(0.,0.)
296         CON=(0.,1.)*BK*ETA/(4.*PI)*R0**2
297         NT=NP-2
298     C
299     C COMPUTE Z(1,LQ) = ZT(LQ)
300     C
301         KQ=1
302         P1=DEL(KQ)/2.
303         P2=PA(KQ)
304         P3=DEL(KQ+1)/2.
305         P4=PA(KQ+1)
306     C
307     C DO THE L LOOP
308     C
309         DO 600 LQ=1,NT
310             PP1=DEL(LQ)/2.
311             PP2=PA(LQ)
312             PP3=DEL(LQ+1)/2.
313             PP4=PA(LQ+1)
314     C
315     C DO THE PHI INTEGRATION
316     C *** FIRST PART FROM PHI(K) TO PHI(K+1)
317     C PHI PRIMED INTEGRATION FOR THE POSITIVE SLOPE OF LQ
318     C
319         SUMA=U0
320         PHA=PA(KQ)
321         DO 100 I=1,NG
322             PHI=P1*XG(I)+P2
323             TK=(PHI-PH(KQ))/DEL(KQ)
324             TKP=1./DEL(KQ)/R0
325             SUM1=U0
326             DO 90 J=1,NG
327                 PHIP=PP1*XG(J)+PP2
328                 TL=(PHIP-PH(LQ))/DEL(LQ)
329                 TLP=1./DEL(LQ)/R0
330                 CC=COS(PHI-PHIP)
331     C
332     C COMPUTE THE MAGNITUDE OF R. NOTE THAT R IS COMPUTED FROM THE
333     C WIRE AXIS TO THE SURFACE OF THE WIRE
334     C
335         RR=R0*SQRT(4.*(SIN((PHI-PHIP)/2.))**2+(A/R0)**2)
336         EXP=CEXP(CMPLX(0.,-BK*RR))/RR
337         SUM1=SUM1+AG(J)*EXP*(TK*TL*CC-TKP*TLP/BK2)
338     90     CONTINUE
339         SUM1=SUM1*PP1
340     C
341     C PHI PRIMED INTEGRATION FOR THE NEGATIVE SLOPE OF LQ
342     C

```



```

343      SUM2=U0
344      DO 80 J=1,NG
345          PHIP=PP3*XG(J)+PP4
346          TL=1.-(PHIP-PH(LQ+1))/DEL(LQ+1)
347          TLP=-1./DEL(LQ+1)/R0
393          CC=COS(PHI-PHIP)
394      C
395      C COMPUTE THE MAGNITUDE OF R. NOTE THAT R IS COMPUTED FROM THE
396      C WIRE AXIS TO THE SURFACE OF THE WIRE
397      C
398          RR=R0*SQRT(4.*(SIN((PHI-PHIP)/2.))**2+(A/R0)**2)
399          EXP=CEXP(CMPLX(0.,-BK*RR))/RR
400          SUM2=SUM2+AG(J)*EXP*(TK*TL*CC-TKP*TLP/BK2)
401      80    CONTINUE
402          SUM2=SUM2*PP3
403          SUMA=SUMA+(SUM1+SUM2)*AG(I)
404      100   CONTINUE
405          SUMA=SUMA*P1
406      C
407      C *** SECOND PART FROM PHI(K+1) TO PHI(K+2)
408      C PHI PRIMED INTEGRATION FOR THE POSITIVE SLOPE OF LQ
409      C
410          SUMB=U0
411          PHA=PA(KQ+1)
412          DO 101 I=1,NG
413              PHI=P3*XG(I)+P4
414              TK=1.-(PHI-PH(KQ+1))/DEL(KQ+1)
415              TKP=-1./DEL(KQ+1)/R0
416              SUM3=U0
417              DO 91 J=1,NG
418                  PHIP=PP1*XG(J)+PP2
419                  TL=(PHIP-PH(LQ))/DEL(LQ)
420                  TLP=1./DEL(LQ)/R0
421                  CC=COS(PHI-PHIP)
422          C
423          C COMPUTE THE MAGNITUDE OF R. NOTE THAT R IS COMPUTED FROM THE
424          C WIRE AXIS TO THE SURFACE OF THE WIRE
425          C
426              RR=R0*SQRT(4.*(SIN((PHI-PHIP)/2.))**2+(A/R0)**2)
427              EXP=CEXP(CMPLX(0.,-BK*RR))/RR
428              SUM3=SUM3+AG(J)*EXP*(TK*TL*CC-TKP*TLP/BK2)
429          91    CONTINUE
430              SUM3=SUM3*PP1
431          C
432          C PHI PRIMED INTEGRATION FOR THE NEGATIVE SLOPE OF LQ
433          C
434              SUM4=U0
435              DO 81 J=1,NG
436                  PHIP=PP3*XG(J)+PP4
437                  TL=1.-(PHIP-PH(LQ+1))/DEL(LQ+1)

```

```

493         TLP=-1./DEL(LQ+1)/R0
494         CC=COS(PHI-PHIP)
495     C
496     C COMPUTE THE MAGNITUDE OF R. NOTE THAT R IS COMPUTED FROM THE
497     C WIRE AXIS TO THE SURFACE OF THE WIRE
498     C
499         RR=R0*SQRT(4.*(SIN((PHI-PHIP)/2.))**2+(A/R0)**2)
500         EXP=CEXP(CMPLX(0.,-BK*RR))/RR
501         SUM4=SUM4+AG(J)*EXP*(TK*TL*CC-TKP*TLP/BK2)
502     81     CONTINUE
503         SUM4=SUM4*PP3
504         SUMB=SUMB+(SUM3+SUM4)*AG(I)
505     101    CONTINUE
506         SUMB=SUMB*P3
507         ZT(LQ)=CON*(SUMA+SUMB)
508         ZMAX=AMAX1(ZMAX,CABS(ZT(LQ)))
509     600    CONTINUE
510         ZT(NT)=ZT(2)
511     C
512     C FILL THE ENTIRE Z MATRIX USING SYMMETRY PROPERTIES
513     C [Z] IS A SYMMETRICAL TOEPLITZ MATRIX
514     C ROW INDEX, I; COL INDEX, J
515     C
516         DO 10 I=1,NT
517             DO 10 J=1,NT
518                 K=(I-1)*NT+J
519                 Z(K)=U0
520                 IJ=IABS(I-J)
521                 IF(IJ.GT.NT) GO TO 10
522                 IJ1=IJ+1
523                 Z(K)=ZT(IJ1)
524     10     CONTINUE
525         RETURN
526     END
527     SUBROUTINE CURVEW(NP,R0,PH,DEL,PA,NG,XG,AG,THR,PHR,R)
528     C
529     C PLANE WAVE EXCITATION VECTOR ELEMENTS FOR A LOOP USING CURVED
530     C BASIS FUNCTIONS. INCIDENCE DIRECTION IS (THR,PHR). THE WIRE LIES
531     C IN THE X-Y PLANE (Z=0).
532     C
533         COMPLEX U0,R(500),SUM1,SUM2,SUM3,SUM4,CEXP,FF
534         COMPLEX GG,CMPLX,SUMT,SUMP
535         DIMENSION PH(500),DEL(500),PA(500),XG(128),AG(128)
536         MT=NP-2
537         U0=(0.,0.)
538         PI=3.14159
539         BK=2.*PI
540         ST=SIN(THR)
541         CT=COS(THR)
542         DO 50 IP=1,MT

```

```

543      SUM1=U0
544      SUM2=U0
545      SUM3=U0
546      SUM4=U0
547      P1=DEL(IP)/2.
548      P2=PA(IP)
549      P3=DEL(IP+1)/2.
550      P4=PA(IP+1)
551      DO 20 I=1,NG
552          PHI=P1*XG(I)+P2
553          CC=COS(PHR-PHI)
554          SS=SIN(PHR-PHI)*CT
555          FF=AG(I)*(PHI-PH(IP))/DEL(IP)*CEXP(CMPLX(0.,BK*R0*ST*CC))
556          SUM1=SUM1+CC*FF
557          SUM2=SUM2+SS*FF
558          PHI=P3*XG(I)+P4
559          CC=COS(PHR-PHI)
560          SS=SIN(PHR-PHI)*CT
561          GG=AG(I)*(1.-(PHI-PH(IP+1))/DEL(IP+1))*CEXP(CMPLX(0.,
562      *   BK*R0*ST*CC))
563          SUM3=SUM3+CC*GG
564          SUM4=SUM4+SS*GG
565      20  CONTINUE
566          SUMP=SUM1*P1+SUM3*P3
567          SUMT=SUM2*P1+SUM4*P3
568      C
569      C R-WIRE-THETA IN R(IP) AND R-WIRE-PHI IN R(IP+MT)
570      C
571          R(IP)=SUMT*R0
572          R(IP+MT)=SUMP*R0
573      50  CONTINUE
574      RETURN
575      END
576

```

```

1      C MAIN PROGRAM *LOOP.FOR*
2      C PLANE WAVE SCATTERING FROM A CIRCULAR LOOP IN THE Z PLANE.
3      C **** RADAR CROSS SECTION CALCULATION ****
4      C
5          COMPLEX Z(15000),B(500),C(500),BP(500),CP(500),U
6          COMPLEX ET,EP,UC,RW(500),U0
7          DIMENSION ECP(400),IPS(500),ANG(400),EDP(400)
8          DIMENSION ZH(200),XT(128),AT(128),XH(200),YH(200)
9          DIMENSION ECV(400),EXV(400),PHC(400),PHX(400)
10         DATA PI/3.14159265/
11         DATA IPRINT/1/
12     C
13     C READ INPUT AND PROGRAM CONTROL PARAMETERS
14     C
15         OPEN(1,FILE='PARAMLST.DAT')
16         READ(1,*)ANGLE,DT,START,STOP,AW,RB,SEG,HARR,GCONST,XLOC,XIPOL
17         LOC=INT(XLOC)
18         IPOL=INT(XIPOL)
19         CLOSE(1)
20     C
21     C READ GAUSSIAN CONSTANTS
22     C
23         OPEN(2,FILE='OUTGLEG')
24         READ(2,*) NT
25         DO 2 I=1,NT
26             READ(2,*) XT(I),AT(I)
27     2   CONTINUE
28         RAD=PI/180.
29         ECX=0.
30         BK=2.*PI
31         ETA=377.
32         U=(0.,1.)
33         U0=(0.,0.)
34         UC=-U*ETA*BK/4./PI
35         CONST=16.*PI**3
36         NT2=NT/2
37     C
38     C OUTLOOP IS THE FILE THAT THE SCATTERED FIELD DATA IS WRITTEN TO
39     C ISTOUT IS THE FILE THAT THE LOOP CURRENT DATA IS WRITTEN TO
40     C
41         OPEN(8,FILE='OUTLOOP.DAT')
42         OPEN(7,FILE='ISTOUT.DAT')
43         DPHI=SEG/RB
44         NP=INT(2.*PI/DPHI)+1
45         DPHI=360./FLOAT(NP)
46         WRITE(6,*) 'DPHI,NP=',DPHI,NP
47     C
48     C GENERATE THE LOOP POINTS. MULTIPLY ALL QUANTITIES BY BK (=2*PI)
49     C
50     C

```

```

51      C CHOOSE THE NUMBER OF POINTS BASED ON THE VALUE OF SEG
52      C
53          AK=AW*BK
54          DO 10 I=1,NP+1
55              PP=FLOAT(I-1)*DPHI*RAD
56              XH(I)=RB*COS(PP)*BK
57              YH(I)=RB*SIN(PP)*BK
58              ZH(I)=0.
59      10  CONTINUE
60          NP=NP+2
61      C
62      C OVERLAP THE ENDS SO THAT THE CURRENT IS CONTINUOUS
63      C
64          XH(NP)=XH(2)
65          YH(NP)=YH(2)
66          ZH(NP)=ZH(2)
67          DO 52 I=1,NP
68              YHB=YH(I)/BK
69              XHB=XH(I)/BK
70              DEL=0.
71              IF(I.NE.1) THEN
72                  DXX=XHB-XH(I-1)/BK
73                  DYY=YHB-YH(I-1)/BK
74                  DEL=SQRT(DXX**2+DYY**2)
75              ENDIF
76      52  CONTINUE
77      C
78      C DEFINE DIMENSIONS OF THE IMPEDANCE MATRIX BLOCKS
79      C
80          MT=NP-2
81          WRITE(6,*) 'MT=',MT
82      C
83      C COMPUTE IMPEDANCE MATRIX ELEMENTS
84      C
85          CALL ZMATWW(1,1,NP,RB,XH,YH,ZH,NT.XT,AT,AK,Z)
86          WRITE(6,*) 'Z COMPUTED'
87          IF(IPRINT.EQ.0) THEN
88              DO 1010 I=1,MT
89                  CZ=CABS(Z(I))
90                  AZ=ATAN2(AIMAG(Z(I)),REAL(Z(I))+1.E-8)/RAD
91                  WRITE(6,*) 'I,Z=',I,Z(I),CZ,CZ/ZMAX,AZ
92      1010  CONTINUE
93          ENDIF
94      C
95      C PERFORM LU DECOMPOSITION
96      C
97          CALL DECOMP(MT,IPS,Z)
98          WRITE(6,*) 'Z DECOMPOSED'
99      C
100     C BEGIN FIELD CALCULATIONS.  PHI FOR PATTERN CUT (DEGREES)=PHID

```

```

101      C
102      PHID=0.
103      PHR0=PHID*RAD
104      IT=INT((STOP-START)/DT)+1
105      WRITE(7,*) IT,MT,0,0
106      DO 500 I=1,IT
107          THETA=FLOAT(I-1)*DT+START
108          THR=THETA*RAD
109          PHR=PHR0
110          IF(THETA.LT.180.) GO TO 99
111          THR=(360.-THETA)*RAD
112          PHR=PHR0+PI
113      99  CONTINUE
114          ET=U0
115          EP=U0
116      C
117      C COMPUTE THE EXCITATION VECTOR
118      C
119          CALL PLANEW(NP,XH,YH,ZH,THR,PHR,RW)
120          IF (LOC .EQ. 0) THEN
121      C
122      C CIRCULAR POLARIZATION IF LOC=1 ELSE LINEAR
123      C
124          IF(IPOL.EQ.1) THEN
125      C
126      C THETA POLARIZED INCIDENT WAVE (IPOL=1)
127      C
128          DO 101 L=1,MT
129      101      B(L)=RW(L)
130          ELSE
131      C
132      C PHI POLARIZED INCIDENT WAVE (IPOL=2)
133      C
134          DO 102 L=1,MT
135      102      B(L)=RW(L+MT)
136          ENDIF
137      C
138      C PERFORM GAUSSIAN ELIMINATION TO DETERMINE [C]
139      C
140          CALL SOLVE(MT,IPS,Z,B,C)
141          DO 210 L=1,MT
142              WRITE(7,*) L,L*DPHI,CABS(C(L)),ATAN2(REAL(C(L)),
143      *  AIMAG(C(L)))/RAD
144              ET=ET+(RW(L)/BK)*C(L)
145      210      EP=EP+(RW(L+MT)/BK)*C(L)
146          ELSE
147      C
148      C THETA PLOARIZED INCIDENT WAVE
149      C
150          DO 221 L=1,MT

```



```

151      221      B(L)=RW(L)
152      C
153      C PHI POLARIZED INCIDENT WAVE
154      C PHASE SHIFT FOR CP IS PI/2.
155      C
156          DO 222 L=1,MT
157      222      BP(L)=RW(L+MT)*CEXP(CMPLX(0.,PI/2.))
158          CALL SOLVE(MT,IPS,Z,B,C)
159          CALL SOLVE(MT,IPS,Z,BP,CP)
160          DO 220 L=1,MT
161              WRITE(7,*) L,L*DPHI,CABS(C(L)+CP(L)),ATAN2(REAL(C(L)+CP(L)),
162      *      AIMAG(C(L)+CP(L)))/RAD
163              ET=ET+RW(L)*C(L)+RW(L)*CP(L)
164      220      EP=EP+RW(L+MT)*C(L)+RW(L+MT)*CP(L)
165          ENDIF
166          ET=UC*ET
167          EP=UC*EP
168      C
169      C E-THETA IS CO-POL; E-PHI IS CROSS-POL
170      C
171          ANG(I)=THETA
172          ECV(I)=CABS(ET)
173          EXV(I)=CABS(EP)
174          ECR=REAL(ET)
175          ECI=AIMAG(ET)
176          EXR=REAL(EP)
177          EXI=AIMAG(EP)
178          PHC(I)=ATAN2(ECI,ECR+1.E-20)/RAD
179          PHX(I)=ATAN2(EXI,EXR+1.E-20)/RAD
180          ECX=AMAX1(ECX,ECV(I),EXV(I))
181      500  CONTINUE
182          DO 600 I=1,IT
183              ECV(I)=AMAX1(ECV(I),1.E-10)
184              EXV(I)=AMAX1(EXV(I),1.E-10)
185              ECP(I)=(ECV(I)/ECX)**2
186              EDP(I)=(EXV(I)/ECX)**2
187              ECP(I)=AMAX1(ECP(I),.00001)
188              EDP(I)=AMAX1(EDP(I),.00001)
189              ECP(I)=10.*ALOG10(ECP(I))
190              EDP(I)=10.*ALOG10(EDP(I))
191      600  CONTINUE
192          SIGMA=(ECX**2)*CABS(UC)/(2.*BK)
193          SIGDB=10.*ALOG10(SIGMA)
194          WRITE(6,*) 'SIGMA, IN DB=',SIGMA,SIGDB
195          DO 9000 L=1,IT
196              WRITE(8,*) ANG(L),ECV(L)
197      9000  CONTINUE
198      900  CONTINUE
199      STOP
200      END

```

```

201         SUBROUTINE SOLVE(N,IPS,UL,B,X)
202     C
203     C SUBROUTINE TO SOLVE SYSTEM OF EQUATIONS WITH COMPLEX
204     COEFFICIENTS.
205     C CALL 'DECOMP' FIRST. (FROM MAUTZ AND HARRINGTON)
206     C
207         COMPLEX UL(50000),B(500),X(500),SUM
208         DIMENSION IPS(500)
209         NP1=N+1
210         IP=IPS(1)
211         X(1)=B(IP)
212         DO 2 I=2,N
213             IP=IPS(I)
214             IPB=IP
215             IM1=I-1
216             SUM=(0.,0.)
217             DO 1 J=1,IM1
218                 SUM=SUM+UL(IP)*X(J)
219             1 IP=IP+N
220             2 X(I)=B(IPB)-SUM
221             K2=N*(N-1)
222             IP=IPS(N)+K2
223             X(N)=X(N)/UL(IP)
224             DO 4 IBACK=2,N
225                 I=NP1-IBACK
226                 K2=K2-N
227                 IPI=IPS(I)+K2
228                 IP1=I+1
229                 SUM=(0.,0.)
230                 IP=IPI
231                 DO 3 J=IP1,N
232                     IP=IP+N
233                     3 SUM=SUM+UL(IP)*X(J)
234                     4 X(I)=(X(I)-SUM)/UL(IPI)
235                 RETURN
236             END
237         SUBROUTINE DECOMP(N,IPS,UL)
238     C
239     C SUBROUTINE TO DECOMPOSE SYSTEM OF EQUATIONS.
240     C FROM MAUTZ AND HARRINGTON.
241     C
242         COMPLEX UL(50000),PIVOT,EM
243         DIMENSION SCL(500),IPS(500)
244         DO 5 I=1,N
245             IPS(I)=I
246             RN=0.
247             J1=I
248             DO 2 J=1,N
249                 ULM=ABS(REAL(UL(J1)))+ABS(AIMAG(UL(J1)))
250                 J1=J1+N

```

```

251      IF(RN-ULM) 1,2,2
252      1 RN=ULM
253      2 CONTINUE
254      SCL(I)=1./RN
255      5 CONTINUE
256      NM1=N-1
257      K2=0
258      DO 17 K=1,NM1
259      BIG=0.
260      DO 11 I=K,N
261      IP=IPS(I)
262      IPK=IP+K2
263      SIZE=(ABS(REAL(UL(IPK))))+ABS(AIMAG(UL(IPK))))*SCL(IP)
264      IF(SIZE-BIG) 11,11,10
265      10 BIG=SIZE
266      IPV=I
267      11 CONTINUE
268      IF(IPV-K) 14,15,14
269      14 J=IPS(K)
270      IPS(K)=IPS(IPV)
271      IPS(IPV)=J
272      15 KPP=IPS(K)+K2
273      PIVOT=UL(KPP)
274      KP1=K+1
275      DO 16 I=KP1,N
276      KP=KPP
277      IP=IPS(I)+K2
278      EM=-UL(IP)/PIVOT
279      18 UL(IP)=-EM
280      DO 16 J=KP1,N
281      IP=IP+N
282      KP=KP+N
283      UL(IP)=UL(IP)+EM*UL(KP)
284      16 CONTINUE
285      K2=K2+N
286      17 CONTINUE
287      RETURN
288      END
289      SUBROUTINEZMATWW(NWIRES,NW1,NW2,RB,XH,YH,ZH,NT,XT,AT,AK,ZZ)
290      C
291      C IMPEDANCE ELEMENTS FOR LINEAR BASIS FUNCTIONS.
292      C
293      COMPLEX CEXP,Z(200),ZZ(15000),CON,CMPLX,EXP
294      COMPLEX U0,SUM,SUM1,SUM2,SUM3,SUM4
295      DIMENSION ZH(200),XT(128),AT(128),XH(200),YH(200),UU(200)
296      DIMENSION D1(200),S1(200),C1(200),ZS1(200)
297      DIMENSION XS1(200),YS1(200)
298      DIMENSION CU(200),SU(200)
299      INTEGER NT,NWIRES,NW1(4),NW2(4),NS(4)
300      PI=3.1415926

```

```

301      PI2=2.*PI
302      ETA=377.
303      BK=PI2
304      U0=(0.,0.)
305      CON=(0.,1.)*BK*ETA/(4.*PI)
306      A=AK
307      C
308      C DEFINE GEOMETRY TERMS FOR THE WIRE. XH,YH,ZH ARE ALL KNOWN.
309      C
310      DO 5 L=1,NWIRES
311      5   NS(L)=NW2(L)-NW1(L)+1
312      NS1=NW2(NWIRES)-NW1(1)
313      NPS=NS1+1
314      NTRIA=NPS-2
315      DO 10 N=2,NPS
316      N0=N-1
317      I=NW1(1)+N-1
318      I2=I-1
319      C
320      C AVERAGE VALUES
321      C
322      ZS1(N0)=.5*(ZH(I)+ZH(I2))
323      XS1(N0)=.5*(XH(I)+XH(I2))
324      YS1(N0)=.5*(YH(I)+YH(I2))
325      DX=XH(I)-XH(I2)
326      DY=YH(I)-YH(I2)
327      D1(N0)=SQRT(DX**2+DY**2)
328      UU(N0)=ATAN2(DY,DX+1.E-5)
329      CU(N0)=COS(UU(N0))
330      SU(N0)=SIN(UU(N0))
331      S1(N0)=DR/D1(N0)
332      C1(N0)=DZ/D1(N0)
333      10 CONTINUE
334      IP=1
335      WRITE(6,*) 'IP=',IP
336      DO 600 JQ=1,NTRIA
337      C
338      C DOING I1
339      C
340      I=IP
341      J=JQ
342      CC=COS(UU(I)-UU(J))
343      TIP=1./D1(I)
344      TJP=1./D1(J)
345      T1=D1(I)/2.
346      T2=D1(J)/2.
347      SUM=U0
348      DO 100 K=1,NT
349      T=T1*XT(K)
350      TI=.5+T/D1(I)

```

```

351      XI=XS1(I)+T*CU(I)
352      YI=YS1(I)+T*SU(I)
353      ZI=ZS1(I)
354      DO 100 L=1,NT
355          TP=T2*XT(L)
356          TJ=.5+TP/D1(J)
357          XJ=XS1(J)+TP*CU(J)
358          YJ=YS1(J)+TP*SU(J)
359          ZJ=ZS1(J)
360          RP=SQRT((XI-XJ)**2+(YI-YJ)**2+A**2)
361          EXP=CEXP(CMPLX(0.,-RP))/RP
362          SUM=SUM+AT(L)*AT(K)*EXP*(TI*TJ*CC-TIP*TJP)
363      100  CONTINUE
364      SUM1=SUM*T1*CON*T2
365      C
366      C DOING I2
367      C
368      J=JQ+1
369      CC=COS(UU(I)-UU(J))
370      TIP=1./D1(I)
371      TJP=-1./D1(J)
372      T1=D1(I)/2.
373      T2=D1(J)/2.
374      SUM=U0
375      DO 101 K=1,NT
376          T=T1*XT(K)
377          TI=.5+T/D1(I)
378          XI=XS1(I)+T*CU(I)
379          YI=YS1(I)+T*SU(I)
380          ZI=ZS1(I)
381          DO 101 L=1,NT
382              TP=T2*XT(L)
383              TJ=.5-TP/D1(J)
384              XJ=XS1(J)+TP*CU(J)
385              YJ=YS1(J)+TP*SU(J)
386              ZJ=ZS1(J)
387              RP=SQRT((XI-XJ)**2+(YI-YJ)**2+A**2)
388              EXP=CEXP(CMPLX(0.,-RP))/RP
389              SUM=SUM+AT(L)*AT(K)*EXP*(TI*TJ*CC-TIP*TJP)
390      101  CONTINUE
391      SUM2=SUM*T1*CON*T2
392      C
393      C DOING I3
394      C
395      I=IP+1
396      J=JQ
397      CC=COS(UU(I)-UU(J))
398      TIP=-1./D1(I)
399      TJP=1./D1(J)
400      T1=D1(I)/2.

```

```

401      T2=D1(J)/2.
402      SUM=U0
403      DO 102 K=1,NT
404          T=T1*XT(K)
405          TI=.5-T/D1(I)
406          XI=XS1(I)+T*CU(I)
407          YI=YS1(I)+T*SU(I)
408          ZI=ZS1(I)
409          DO 102 L=1,NT
410              TP=T2*XT(L)
411              TJ=.5+TP/D1(J)
412              XJ=XS1(J)+TP*CU(J)
413              YJ=YS1(J)+TP*SU(J)
414              ZJ=ZS1(J)
415              RP=SQRT((XI-XJ)**2+(YI-YJ)**2+A**2)
416              EXP=CEXP(CMPLX(0.,-RP))/RP
417              SUM=SUM+AT(L)*AT(K)*EXP*(TI*TJ*CC-TIP*TJP)
418      102 CONTINUE
419      SUM3=SUM*T1*CON*T2
420      C
421      C DOING I4
422      C
423          J=JQ+1
424          CC=COS(UU(I)-UU(J))
425          TIP=-1./D1(I)
426          TJP=-1./D1(J)
427          T1=D1(I)/2.
428          T2=D1(J)/2.
429          SUM=U0
430          DO 103 K=1,NT
431              T=T1*XT(K)
432              TI=.5-T/D1(I)
433              XI=XS1(I)+T*CU(I)
434              YI=YS1(I)+T*SU(I)
435              ZI=ZS1(I)
436              DO 103 L=1,NT
437                  TP=T2*XT(L)
438                  TJ=.5-TP/D1(J)
439                  XJ=XS1(J)+TP*CU(J)
440                  YJ=YS1(J)+TP*SU(J)
441                  ZJ=ZS1(J)
442                  RP=SQRT((XI-XJ)**2+(YI-YJ)**2+A**2)
443                  EXP=CEXP(CMPLX(0.,-RP))/RP
444                  SUM=SUM+AT(L)*AT(K)*EXP*(TI*TJ*CC-TIP*TJP)
445      103 CONTINUE
446      SUM4=SUM*T1*CON*T2
447      C
448      C IMPEDANCE ELEMENT FOR IP,JQ
449      C
450      KK=(JQ-1)*NTRIA+IP

```



```

451          Z(JQ)=(SUM1 + SUM2 + SUM3 + SUM4)
452 600  CONTINUE
453      Z(NTRIA)=Z(2)
454  C
455  C FILL THE ENTIRE Z MATRIX USING SYMMETRY PROPERTIES
456  C ROW INDEX, I; COL INDEX, J
457  C
458      DO 12 I=1,NTRIA
459          DO 12 J=1,NTRIA
460              K=(I-1)*NTRIA + J
461              ZZ(K)=U0
462              IJ=IABS(I-J)
463              IF(IJ.GT.NTRIA) GO TO 12
464              IJ1=IJ + 1
465              ZZ(K)=Z(IJ1)
466 12  CONTINUE
467      CLOSE(2)
468      RETURN
469      END
470      SUBROUTINE PLANEW(NP,XH,YH,ZH,THR,PHR,R)
471  C
472  C PLANE WAVE EXCITATION VECTOR ELEMENTS FOR WIRE AND
473  C INCIDENCE DIRECTION IS (THR.PHR).
474  C WIRE LIES IN THE X-Y PLANE (Z=0)
475  C
476      COMPLEX U0,C,R(2000),CEXP,EXP,F11,F12,S1,D1,CMPLX
477      DIMENSION ZH(500),XH(500),YH(500)
478      MP2=NP-1
479      MT2=NP-2
480      U0=(0.,0.)
481      CC=COS(THR)
482      SS=SIN(THR)
483      CP=COS(PHR)
484      SP=SIN(PHR)
485      UP=SS*CP
486      VP=SS*SP
487      DO 12 IP=1,MP2
488          II=IP
489          I=II + 1
490          ZS=.5*(ZH(I) + ZH(II))
491          XS=.5*(XH(I) + XH(II))
492          YS=.5*(YH(I) + YH(II))
493          DX=XH(I)-XH(II)
494          DY=YH(I)-YH(II)
495          D1=SQRT(DX**2 + DY**2)
496          SU=DY/D1
497          CU=DX/D1
498  C FOR WIRES IN THE XY PLANE SIN(V)=1 AND COS(V)=0
499          SV=1.0
500          CV=0.0

```

```

501      C
502      C WIRE SEGMENT CALCULATIONS
503      C
504          A=UP*CU+VP*SU
505          B=UP*XS+VP*YS
506          C=CMPLX(0.,A)
507          EXP=CEXP(CMPLX(0.,B))
508          AA=CC*(CU*CP+SU*SP)
509          BB=SU*CP-SP*CU
510          PSI=D1*A/2.
511          IF(PSI.NE.0.) GO TO 60
512          SINC=1.
513          GO TO 61
514      60   SINC=SIN(PSI)/PSI
515      61   COSP=COS(PSI)
516          FI1=SINC*D1*EXP/2.
517          FI2=(0.,0.)
518          IF(ABS(A).LT.1.E-4) GO TO 62
519          CSP=COSP-SINC
520          IF(ABS(CSP).LT.1.E-4) GO TO 62
521          FI2=EXP/C*CSP
522      62   CONTINUE
523          SI=FI1+FI2
524          DI=FI1-FI2
525      C
526      C R-WIRE-THETA
527      C
528          IF(IP.EQ.MP2) GO TO 10
529          R(IP)=AA*SI
530          R(IP+MT2)=BB*SI
531      10   CONTINUE
532      C
533      C R-WIRE-PHI
534      C
535          14 IF(IP.EQ.1) GO TO 12
536          R(IP-1)=R(IP-1)+AA*DI
537          R(IP-1+MT2)=R(IP-1+MT2)+BB*DI
538          12 CONTINUE
539      210 RETURN
540          END
541

```

```

1      C MAIN PROGRAM *HARLOOP.F*
2      C PLANE WAVE SCATTERING FROM A CIRCULAR LOOP IN THE Z PLANE.
3      C **** RADAR CROSS SECTION CALCULATION ****
4      C USING HARRINGTON'S FORMULATION FROM THE BOOK 'FIELD COMP. BY
5      C MM' (P.83 TO 95)
6      C
7      COMPLEX Z(5000),E(250),C(250),EX,EC,ET,EP,RW(1000),UC,EPhi(500)
8      COMPLEX CPhi(500)
9      DIMENSION ECP(500),IPS(250),ANG(500),EDP(500),XT(300),AT(300)
10     DIMENSION ECV(500),EXV(500),PHC(500),PHX(500)
11     DATA PI/3.14159265/
12     DATA IPRINT/1/,ITEST/1/
13     RAD=PI/180.
14     ECX=0.
15     BK=2.*PI
16     ETA=377.
17     UC=(0,-1)*ETA*BK/(4.*PI)
18     PHIRD=0.
19     OPEN(1,FILE='PARAMLST.DAT')
20     READ(1,*) ANGLE,DT,START,STOP,A,B,SEG,AHARR,GCONST,XLOC,XIPOL
21     IPOL=INT(XIPOL)
22     LOC=INT(XLOC)
23     CLOSE(1)
24     NM=INT(AHARR)
25     CON=(377.*BK)**2/2./BK
26     OPEN(2,FILE='OUTGLEG')
27     READ(2,*) NT
28     DO 1 I=1,NT
29         READ(2,*) XT(I),AT(I)
30     1 CONTINUE
31     OPEN(8,FILE='OUTHARR.DAT')
32     OPEN(7,FILE='IHARROUT.DAT')
33     NROW=2*NM+1
34     WRITE(6,1300) B,A,NROW,NT
35     1300 FORMAT(//,5X,'PLANE WAVE SCATTERING BY A CIRCULAR LOOP',/
36     * ,5X, 'USING METHOD IN HARRINGTONS MM TEXT BOOK',/
37     * ,5X,'LOOP RADIUS (WAVL)=' ,F8.4,/,5X,'WIRE RADIUS (WAVL)=' ,
38     * F8.4,/,5X,'NUMBER OF AZIMUTHAL MODES (INCLUDING ZERO)=' ,I3,
39     * /,5X,'NUMBER OF INTEGRATION POINTS IN PHI=' ,I4)
40     C
41     C COMPUTE IMPEDANCE MATRIX ELEMENTS
42     C
43     WRITE(6,*) 'CALLING ZMAT'
44     CALL ZMATWW(NM,A,B,NT,XT,AT,Z)
45     WRITE(6,*) 'WIRE IMPEDANCE COMPUTED'
46     CALL DECOMP(NROW,IPS,Z)
47     WRITE(6,*) 'Z DECOMPOSED'
48     C
49     C BEGIN FIELD CALCULATIONS
50     C

```

```

51      PHR0=PHIRD*RAD
52      IT=INT((STOP-START)/DT)+1
53      WRITE(7,*) IT,NROW
54      DO 500 I=1,IT
55          THETA=FLOAT(I-1)*DT+START
56          WRITE(6,*) 'THETA=',THETA
57          THR=THETA*RAD
58          PHR=PHR0
59          IF(THETA.LT.180.) GO TO 99
60          THR=(360.-THETA)*RAD
61          PHR=PHR0+PI
62      99  CONTINUE
63          ET=(0.,0.)
64          EP=(0.,0.)
65          CALL PLANEW(NM,B,THR,PHR,RW)
66      C TRANSMIT VECTOR ELEMENTS ARE TRANSPOSED FORMS OF RECEIVE
67      VECTOR
68      C ALSO THE THETA COMPONENT GETS A NEGATIVE SIGN
69          IF(LOC .EQ. 0) THEN
70              IF(IPOL.EQ.1) THEN
71                  C
72                  C THETA POLARIZED INCIDENT WAVE (IPOL=1)
73                  C
74                      DO 101 L=1,NROW
75      101          E(NROW-L+1)=RW(L)
76                      ELSE
77                  C
78                  C PHI POLARIZED INCIDENT WAVE (IPOL=2)
79                  C
80                      DO 102 L=1,NROW
81      102          E(NROW-L+1)=RW(L+NROW)
82                      ENDIF
83                      WRITE(6,*) 'CALLING SOLVE'
84                      CALL SOLVE(NROW,IPS,Z,E,C)
85                      WRITE(6,*) 'RETURNED FROM SOLVE'
86                      DO 210 L=1,NROW
87                          WRITE(7,*) C(L)
88                          ET=ET+RW(L)*C(L)
89      210          EP=EP+RW(L+NROW)*C(L)
90                      ELSE
91                  C
92                  C E-THETA IS CO-POL; E-PHI IS CROSS-POL
93                  C
94                      DO 221 L=1,NROW
95      221          E(NROW-L+1)=RW(L)
96                      DO 222 L=1,NROW
97      222          EPHI(NROW-L+1)=RW(L+NROW)*CEXP(CMPLX(0.,PI/2.))
98                      CALL SOLVE(NROW,IPS,Z,E,C)
99                      CALL SOLVE(NROW,IPS,Z,EPHI,CPhi)
100                     DO 220 L=1,NROW

```

```

101         WRITE(7,*) C(L)+CPHI(L)
102         ET=ET+RW(L)*C(L)+RW(L)*CPHI(L)
103     220     EP=EP+RW(L+NROW)*C(L)+RW(L+NROW)*CPHI(L)
104         ENDIF
105         EC=UC*ET
106         EX=UC*EP
107         ANG(I)=THETA
108         ECV(I)=CABS(EC)
109         EXV(I)=CABS(EX)
110         ECR=REAL(EC)
111         ECI=AIMAG(EC)
112         EXR=REAL(EX)
113         EXI=AIMAG(EX)
114         PHC(I)=ATAN2(ECI,ECR+1.E-20)/RAD
115         PHX(I)=ATAN2(EXI,EXR+1.E-20)/RAD
116         ECX=AMAX1(ECX,ECV(I),EXV(I))
117     500 CONTINUE
118         DO 600 I=1,IT
119             ECV(I)=AMAX1(ECV(I),1.E-10)
120             EXV(I)=AMAX1(EXV(I),1.E-10)
121             ECP(I)=(ECV(I)/ECX)**2
122             EDP(I)=(EXV(I)/ECX)**2
123             ECP(I)=AMAX1(ECP(I),.00001)
124             EDP(I)=AMAX1(EDP(I),.00001)
125             ECP(I)=10.*ALOG10(ECP(I))
126             EDP(I)=10.*ALOG10(EDP(I))
127     600 CONTINUE
128         SIGMA=(ECX**2)*4.*PI
129         SIGDB=10.*ALOG10(SIGMA)
130         WRITE(6,*) 'BACKSCATTER, IN DB=',SIGMA,SIGDB
131         OPEN(2,FILE='RCS.DAT')
132         WRITE(2,*) SIGMA,SIGDB
133         CLOSE(2)
134     C
135     C PRINT FIELD POINTS
136     C
137         DO 9000 L=1,IT
138             WRITE(8,*) ANG(L),ECV(L)
139     5016     FORMAT(5X,F6.1,3X,2(F8.4,3X,F7.1,3X,F7.2,3X))
140     9000 CONTINUE
141     900 CONTINUE
142     9998 STOP
143     END
144     SUBROUTINE SOLVE(N,IPS,UL,B,X)
145     COMPLEX UL(5000),B(250),X(250),SUM
146     DIMENSION IPS(250)
147     NP1=N+1
148     IP=IPS(1)
149     X(1)=B(IP)
150     DO 2 I=2,N

```

```

151      IP=IPS(I)
152      IPB=IP
153      IM1=I-1
154      SUM=(0.,0.)
155      DO 1 J=1,IM1
156      SUM=SUM+UL(IP)*X(J)
157      1 IP=IP+N
158      2 X(I)=B(IPB)-SUM
159      K2=N*(N-1)
160      IP=IPS(N)+K2
161      X(N)=X(N)/UL(IP)
162      DO 4 IBACK=2,N
163      I=NP1-IBACK
164      K2=K2-N
165      IPI=IPS(I)+K2
166      IP1=I+1
167      SUM=(0.,0.)
168      IP=IPI
169      DO 3 J=IP1,N
170      IP=IP+N
171      3 SUM=SUM+UL(IP)*X(J)
172      4 X(I)=(X(I)-SUM)/UL(IPI)
173      RETURN
174      END
175      SUBROUTINE DECOMP(N,IPS,UL)
176      COMPLEX UL(5000),PIVOT,EM
177      DIMENSION SCL(250),IPS(250)
178      DO 5 I=1,N
179      IPS(I)=I
180      RN=0.
181      J1=I
182      DO 2 J=1,N
183      ULM=ABS(REAL(UL(J1)))+ABS(AIMAG(UL(J1)))
184      J1=J1+N
185      IF(RN-ULM) 1,2,2
186      1 RN=ULM
187      2 CONTINUE
188      SCL(I)=1./RN
189      5 CONTINUE
190      NM1=N-1
191      K2=0
192      DO 17 K=1,NM1
193      BIG=0.
194      DO 11 I=K,N
195      IP=IPS(I)
196      IPK=IP+K2
197      SIZE=(ABS(REAL(UL(IPK)))+ABS(AIMAG(UL(IPK))))*SCL(IP)
198      IF(SIZE-BIG) 11,11,10
199      10 BIG=SIZE
200      IPV=I

```



```

201      11 CONTINUE
202      IF(IPV-K) 14,15,14
203      14 J=IPS(K)
204      IPS(K)=IPS(IPV)
205      IPS(IPV)=J
206      15 KPP=IPS(K)+K2
207      PIVOT=UL(KPP)
208      KP1=K+1
209      DO 16 I=KP1,N
210      KP=KPP
211      IP=IPS(I)+K2
212      EM=-UL(IP)/PIVOT
213      18 UL(IP)=-EM
214      DO 16 J=KP1,N
215      IP=IP+N
216      KP=KP+N
217      UL(IP)=UL(IP)+EM*UL(KP)
218      16 CONTINUE
219      K2=K2+N
220      17 CONTINUE
221      RETURN
222      END
223      SUBROUTINE ZMATWW(NM,A,B,NT,XT,AT,Z)
224      C
225      C *** MODS FOR LOOP -- USING HARRINGTON'S TEXT BOOK EQUATIONS AS A
226      C CHECK FOR OTHER METHODS. NM IS THE NUMBER OF AZIMUTHAL MODES
227      C USED
228      C
229      COMPLEX Z(5000),CON,CMPLX,FK
230      DIMENSION XT(300),AT(300)
231      PI=3.1415926
232      BK=2.*PI
233      CON=CMPLX(0.,PI*377.*BK*B)
234      NROW=2.*NM+1
235      DO 10 I=-NM,NM
236      DO 10 J=-NM,NM
237      IJ=J+NM+1+(I+NM)*NROW
238      Z(IJ)=(0.,0.)
239      10 CONTINUE
240      C
241      C ONLY DIAGONAL ELEMENTS ARE NONZERO. ALTHOUGH SYMMETRY EXISTS
242      C BETWEEN
243      C Z(-N,-N) AND Z(N,N) IT IS NOT BEING USED.
244      C
245      DO 20 I=-NM,NM
246      J=I+NM+1+(I+NM)*NROW
247      IP=I+1
248      IM=I-1
249      Z(J)=(.5*FK(IM,B,A,NT,XT,AT)+.5*FK(IP,B,A,NT,XT,AT)-
250      * (I/BK/B)**2*FK(I,B,A,NT,XT,AT))*CON

```

```

251      20  CONTINUE
252      RETURN
253      END
254      SUBROUTINE PLANEW(N,B,THR,PHR,R)
255      C
256      C PLANE WAVE RECEIVE VECTOR ELEMENTS FOR WIRE USING THE
257      C FORMULATION FROM HARRINGTON'S BOOK. N IS THE NUMBER OF
258      C AZIMUTHAL MODES. NOTE THAT B(N)=R(-N) (B IS EXCITATION AND
259      C R IS RECEIVE).
260      C
261      COMPLEX R(1000),CEXP,EXP,CMPLX
262      PI=3.14159
263      BK=2.*PI
264      CT=COS(THR)
265      ST=SIN(THR)
266      RR=BK*B*ST
267      NROW=2*N+1
268      C DO THETA RECEIVE COMPONENTS FIRST
269      DO 10 I=-N,N
270          IP=I+1
271          IM=I-1
272          EXP=CEXP(CMPLX(0.,I*PHR))
273          R(I+N+1)=-PI*B*(0.,1.)**I*EXP*(BESSJ(IP,RR)+BESSJ(IM,RR))*CT
274      10  CONTINUE
275      C NOW DO PHI RECEIVE COMPONENTS
276      DO 20 I=-N,N
277          IP=I+1
278          IM=I-1
279          EXP=CEXP(CMPLX(0.,I*PHR))
280          R(I+N+1+NROW)=PI*B*(0.,1.)**(I+1)*EXP*(BESSJ(IP,RR)
281          * -BESSJ(IM,RR))
282      20  CONTINUE
283      RETURN
284      END
285      COMPLEX FUNCTION FK(N,B,A,NT,XT,AT)
286      COMPLEX SUM,EXP1,EXP2,CEXP,CMPLX
287      DIMENSION XT(300),AT(300)
288      PI=3.14159
289      BK=2.*PI
290      CC=1./BK
291      P1=(2.*PI-0.)/2.
292      P2=(2.*PI+0.)/2.
293      SUM=(0.,0.)
294      DO 10 I=1,NT
295          PHI=P1*XT(I)+P2
296          RR=SQRT(4.*SIN(PHI/2.)**2+(A/B)**2)
297          EXP1=CEXP(CMPLX(0.,-BK*B*RR))
298          EXP2=CEXP(CMPLX(0.,-N*PHI))
299          SUM=SUM+AT(I)*EXP1*EXP2/RR
300      10  CONTINUE

```

```

301         FK=SUM*P1*CC
302         RETURN
303         END
304         FUNCTION BESSJ(NN,X)
305 C RETURNS THE BESSEL FUNCTION B OF ORDER N (> 1) AND REAL
306 C ARGUMENT X.
307         PARAMETER (IACC=40,BIGNO=1.E10,BIGNI=1.E-10)
308         IF(NN.LT.0) N=-NN
309         IF(NN.GE.0) N=NN
310         KC=3
311         IF(N.EQ.0) KC=1
312         IF(N.EQ.1) KC=2
313         GO TO (1,2,3),KC
314 1      BESSJ=BESSJ0(X)
315         GO TO 4
316 2      BESSJ=BESSJ1(X)
317         GO TO 4
318 3      BESSJ=0.
319         IF(ABS(X).LT.1.E-5) GO TO 4
320         TOX=2./X
321         IF(X.GT.FLOAT(N)) THEN
322             BJM=BESSJ0(X)
323             BJ=BESSJ1(X)
324             DO 11 J=1,N-1
325                 BJP=J*TOX*BJ-BJM
326                 BJM=BJ
327                 BJ=BJP
328 11      CONTINUE
329             BESSJ=BJ
330         ELSE
331             M=2*((N+INT(SQRT(FLOAT(IACC*N)))))/2)
332             BESSJ=0.
333             JSUM=0.
334             SUM=0.
335             BJP=0.
336             BJ=1.
337             DO 12 J=M,1,-1
338                 BJM=J*TOX*BJ-BJP
339                 BJP=BJ
340                 BJ=BJM
341             IF(ABS(BJ).GT.BIGNO) THEN
342                 BJ=BJ*BIGNI
343                 BJP=BJP*BIGNI
344                 BESSJ=BESSJ*BIGNI
345                 SUM=SUM*BIGNI
346             ENDIF
347             IF(JSUM.NE.0) SUM=SUM+BJ
348             JSUM=1-JSUM
349             IF(J.EQ.N) BESSJ=BJP
350 12      CONTINUE

```

```

351         SUM=2.*SUM-BJ
352         BESSJ=BESSJ/SUM
353     ENDIF
354 4     CONTINUE
355     IF(NN.LT.0) BESSJ=(-1.)*N*BESSJ
356     RETURN
357     END
358     FUNCTION BESSJ0(X)
359 C
360 C BESSEL FUNCTION OF 0 ORDER, REAL ARGUMENT X
361 C (SEE 'NUMERICAL RECIPES', P.172)
362 C
363     REAL*8 Y,P1,P2,P3,P4,P5,Q1,Q2,Q3,Q4,Q5,R1,R2,R3,R4,R5,R6,
364     * S1,S2,S3,S4,S5,S6
365     DATA P1,P2,P3,P4,P5/1.D0,-.109862827D-2,.2734510407D-4,
366     * -.2073370639D-5,.2093887211D-6/
367     DATA Q1,Q2,Q3,Q4,Q5/-.1562499995D-1,.1430488765D-3,
368     * -.6911147651D-5,.7621095161D-6,-.934945152D-7/
369     DATA R1,R2,R3,R4,R5,R6/57568490574.D0,-13362590354.D0,
370     * 651619640.7D0,-11214424.18D0,77392.33017D0,-184.9052456D0/
371     DATA S1,S2,S3,S4,S5,S6/57568490411.D0,1029532985.D0,
372     * 9494680.718D0,59272.64853D0,267.8532712D0,1.D0/
373     IF(ABS(X).LT.8.) THEN
374         Y=X**2
375         BESSJ0=(R1 + Y*(R2 + Y*(R3 + Y*(R4 + Y*(R5 + Y*R6))))/
376     * (S1 + Y*(S2 + Y*(S3 + Y*(S4 + Y*(S5 + Y*S6))))
377     ELSE
378         AX=ABS(X)
379         Z=8./AX
380         Y=Z**2
381         XX=AX-.785398164
382         BESSJ0=SQRT(.636619772/AX)*(COS(XX)*(P1 + Y*(P2 + Y*(P3 +
383     * Y*(P4 + Y*P5))))-Z*SIN(XX)*(Q1 + Y*(Q2 + Y*(Q3 +
384     * Y*(Q4 + Y*Q5))))
385     ENDIF
386     RETURN
387     END
388     FUNCTION BESSJ1(X)
389 C
390 C BESSEL FUNCTION B OF ORDER 1, REAL ARGUMENT X
391 C (SEE 'NUMERICAL RECIPES', P.173)
392 C
393     REAL*8 Y,P1,P2,P3,P4,P5,Q1,Q2,Q3,Q4,Q5,R1,R2,R3,R4,R5,R6,
394     * S1,S2,S3,S4,S5,S6
395     DATA P1,P2,P3,P4,P5/1.D0,.183105D-2,-.3516396496D-4,
396     * .2457520174D-5,-.20337019D-6/
397     DATA Q1,Q2,Q3,Q4,Q5/.04687499995D0,-.2002690873D-3,
398     * .8449199096D-5,-.99228987D-6,.105787412D-6/
399     DATA R1,R2,R3,R4,R5,R6/72362614232.D0,-7895059235.D0,
400     * 242396853.1D0,-2972611.439D0,15704.4826D0,-30.16036606D0/

```

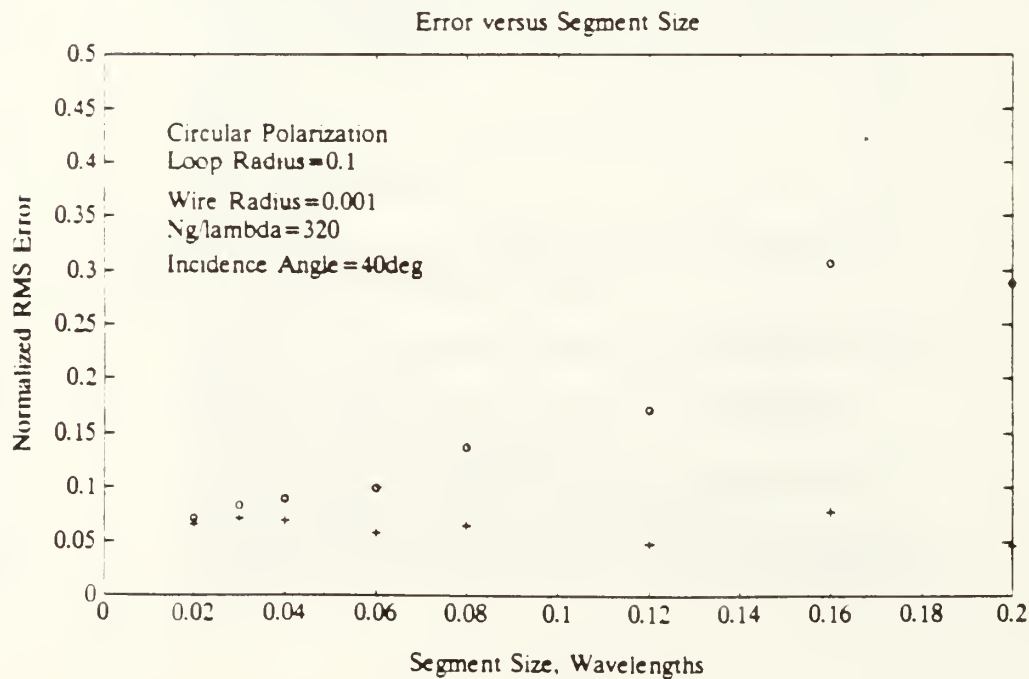
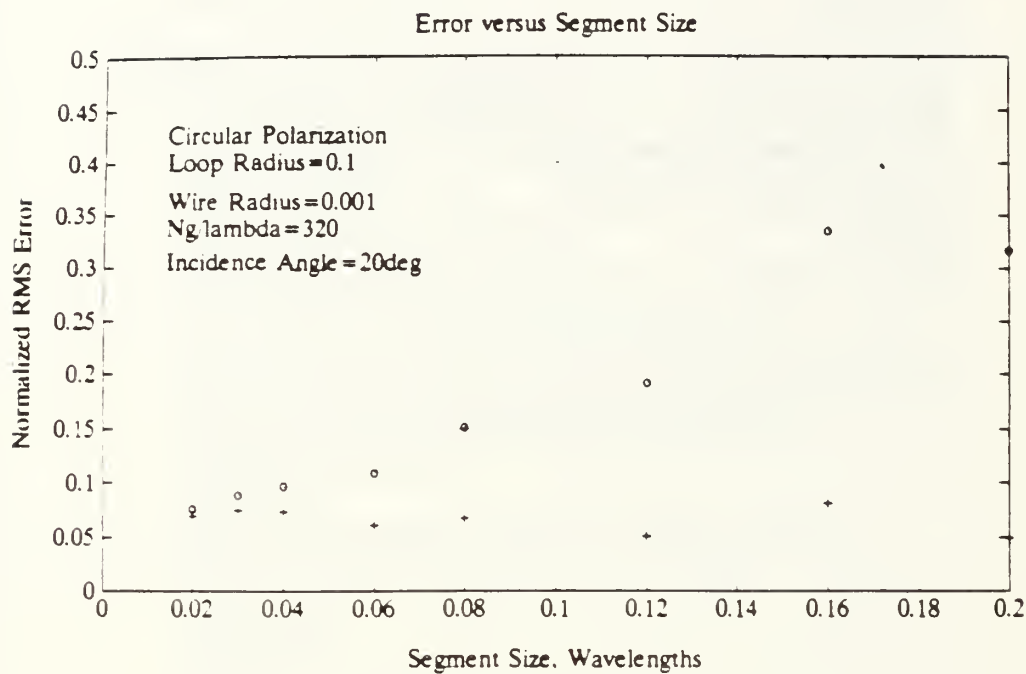
```

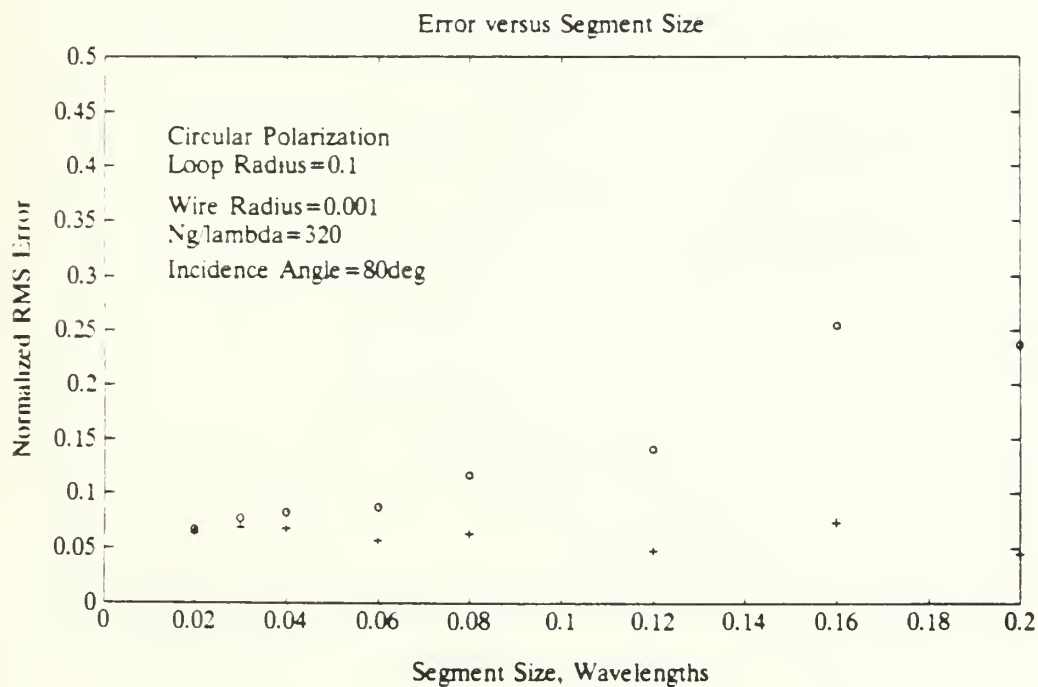
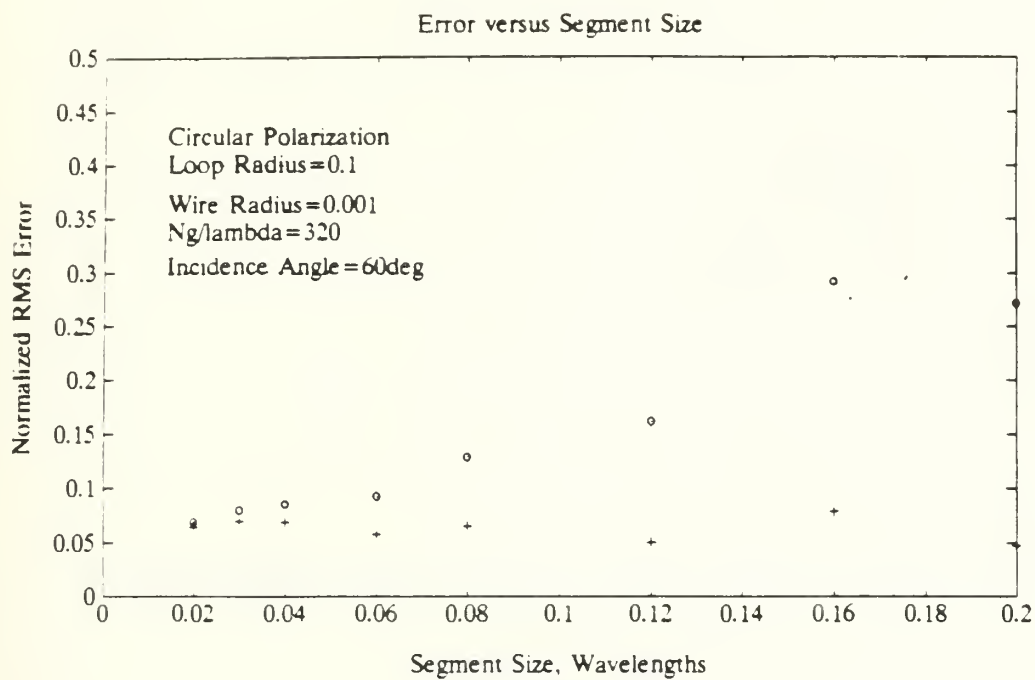
401 DATA S1,S2,S3,S4,S5,S6/144725228442.D0,2300535178.D0,
402 * 18583304.74D0,99447.43394D0,376.9991397D0,1.D0/
403 IF(ABS(X).LT.8.) THEN
404 Y=X**2
405 BESSJ1=X*(R1+Y*(R2+Y*(R3+Y*(R4+Y*(R5+Y*R6)))))/
406 * (S1+Y*(S2+Y*(S3+Y*(S4+Y*(S5+Y*S6))))
407 ELSE
408 AX=ABS(X)
409 Z=8./AX
410 Y=Z**2
411 XX=AX-2.356194491
412 BESSJ1=SQRT(.636619772/AX)*(COS(XX)*(P1+Y*(P2+Y*(P3+
413 * Y*(P4+Y*P5))))-Z*SIN(XX)*(Q1+Y*(Q2+Y*(Q3+
414 * Y*(Q4+Y*Q5)))))*SIGN(1.,X)
415 ENDIF
416 RETURN
417 END

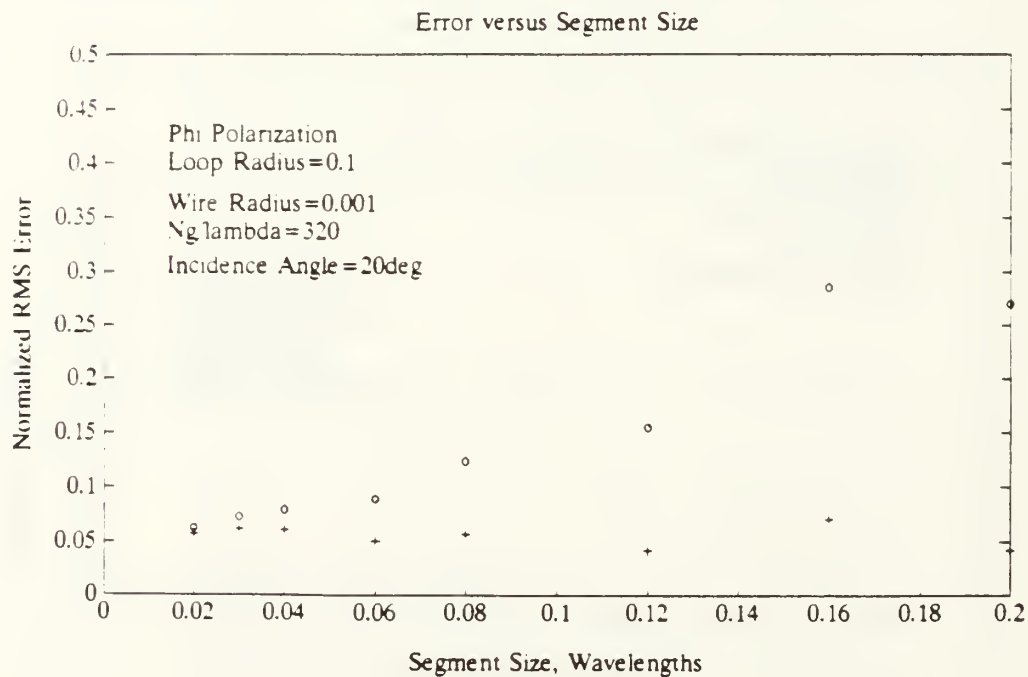
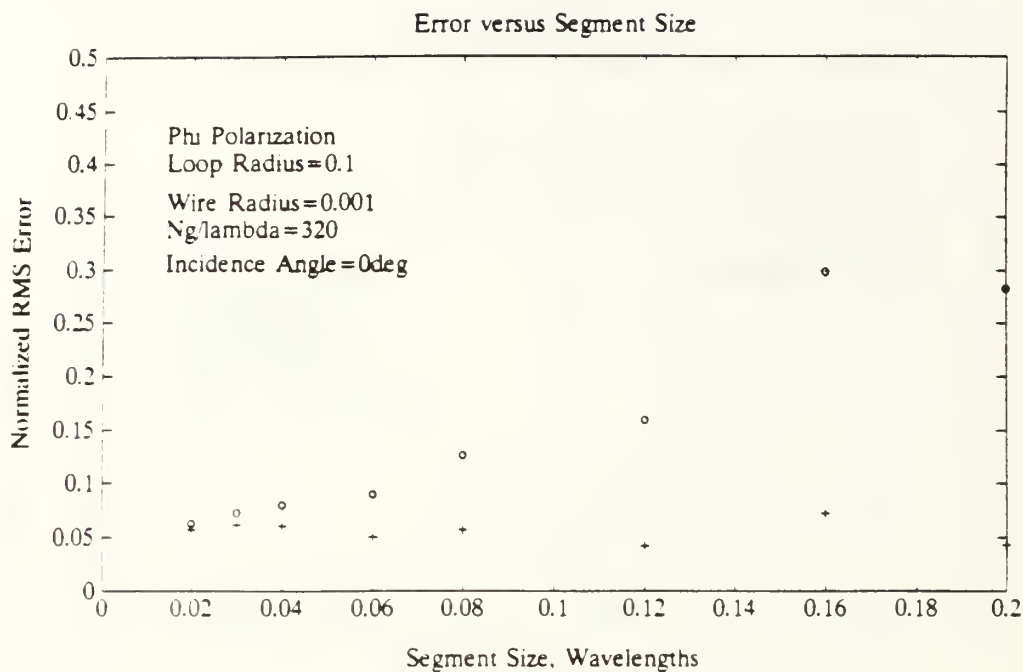
```

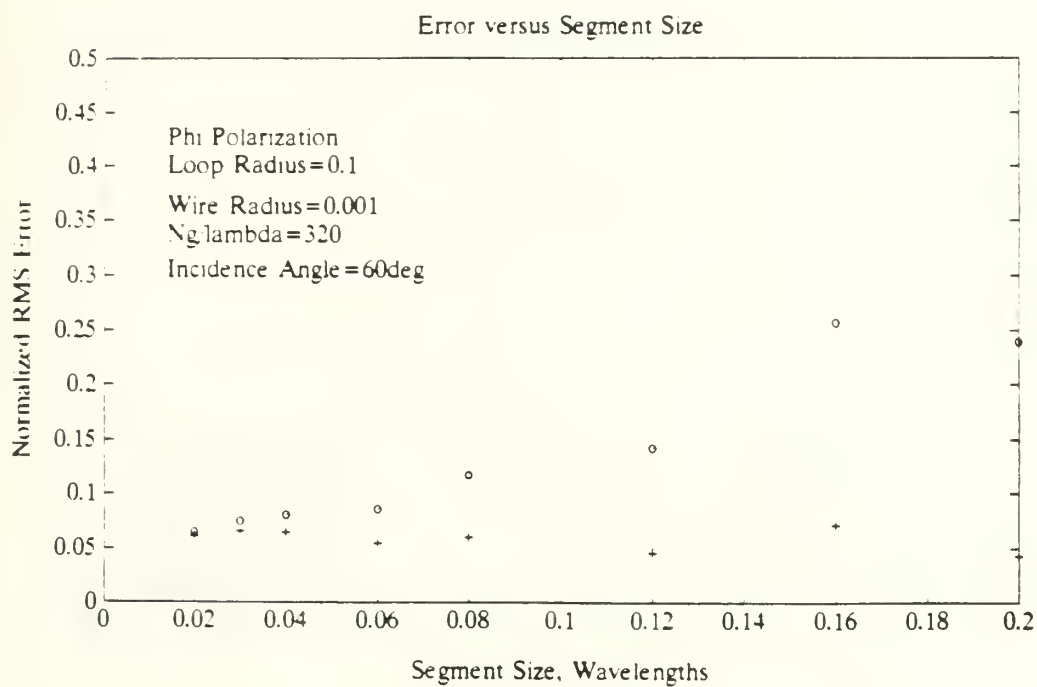
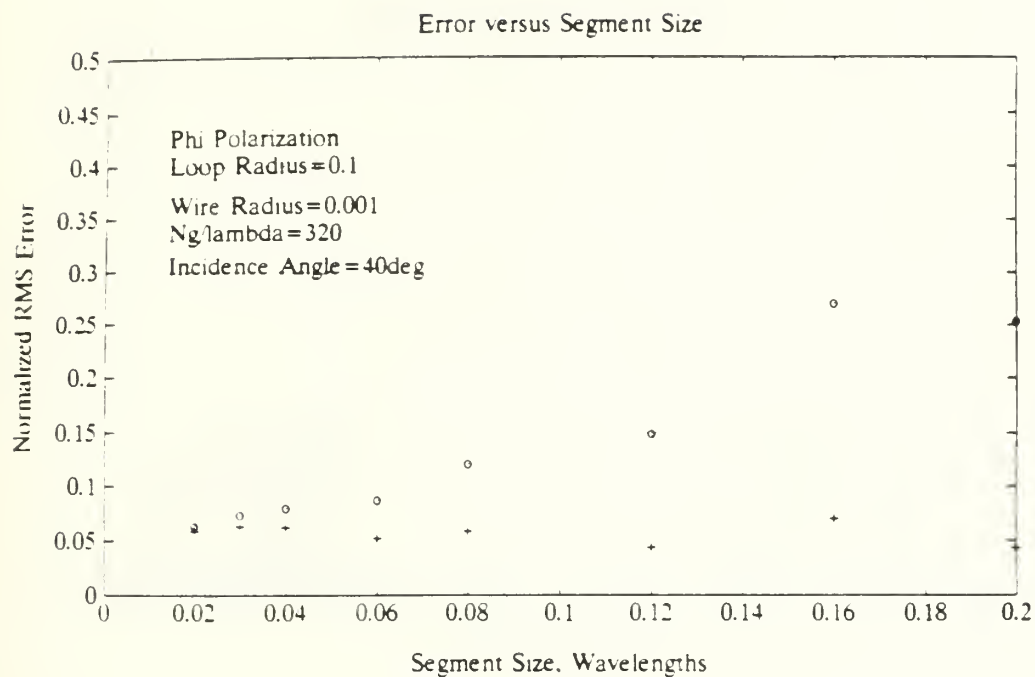
APPENDIX B

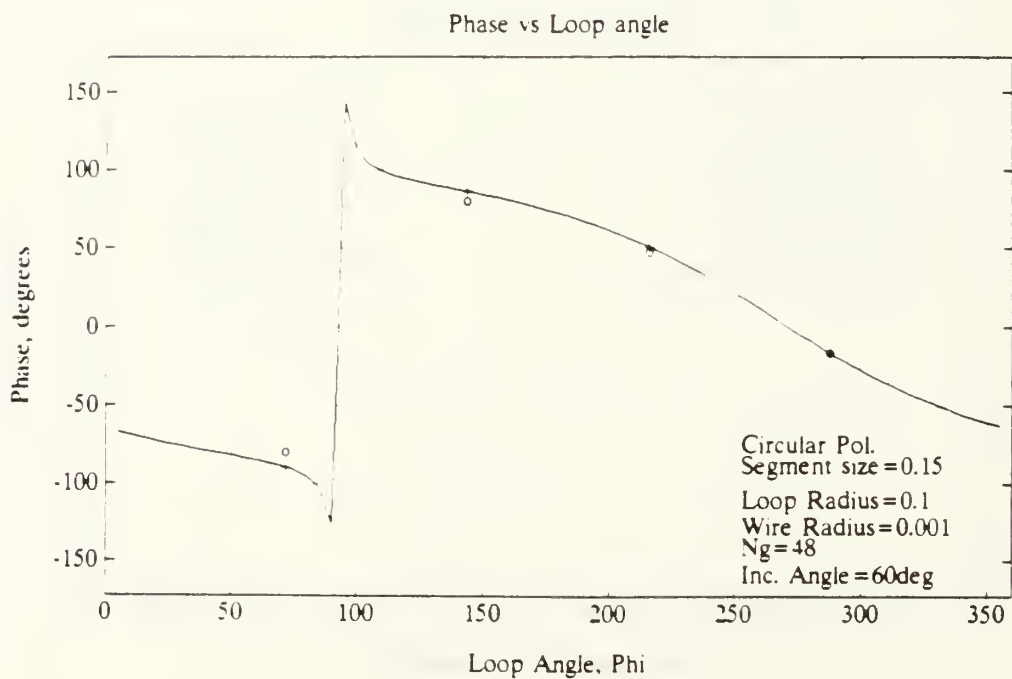
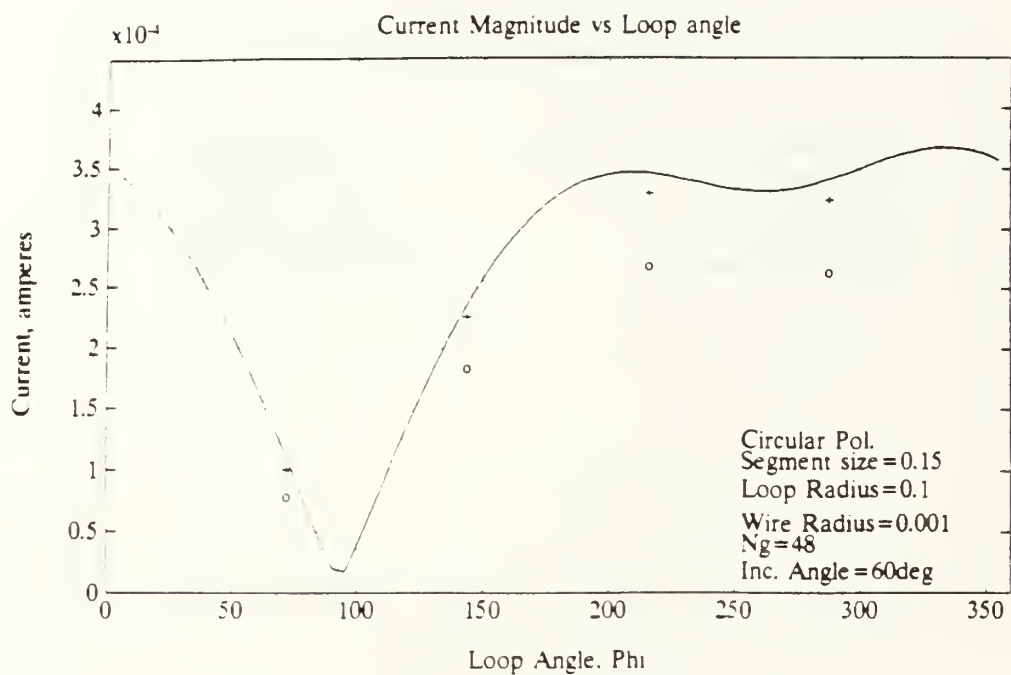
ADDITIONAL PLOTS OF CURRENT AND RMS ERROR

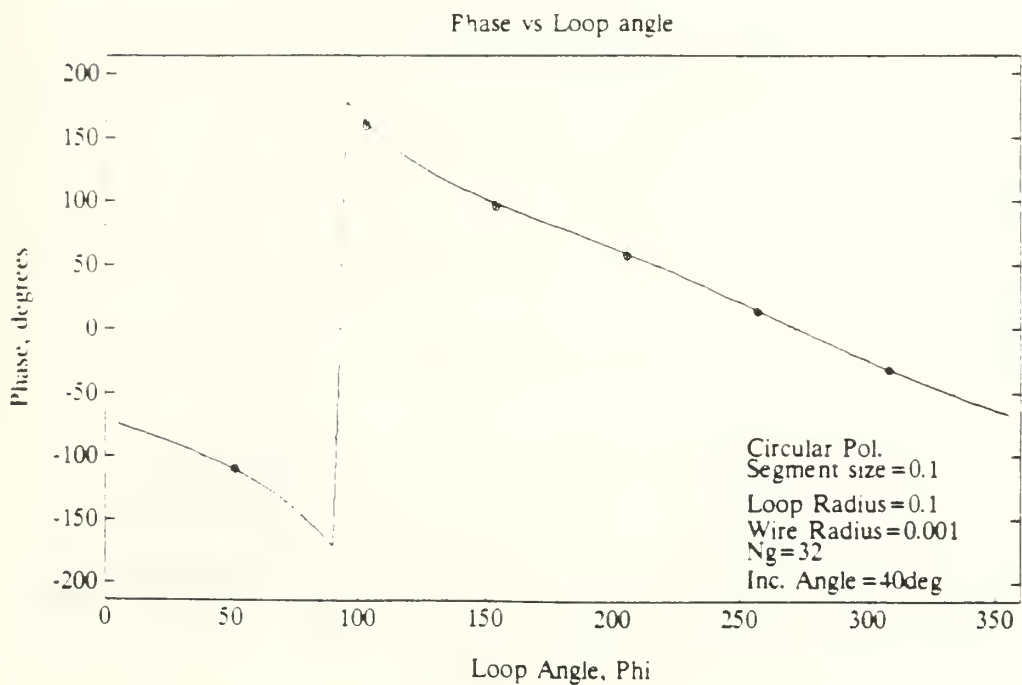
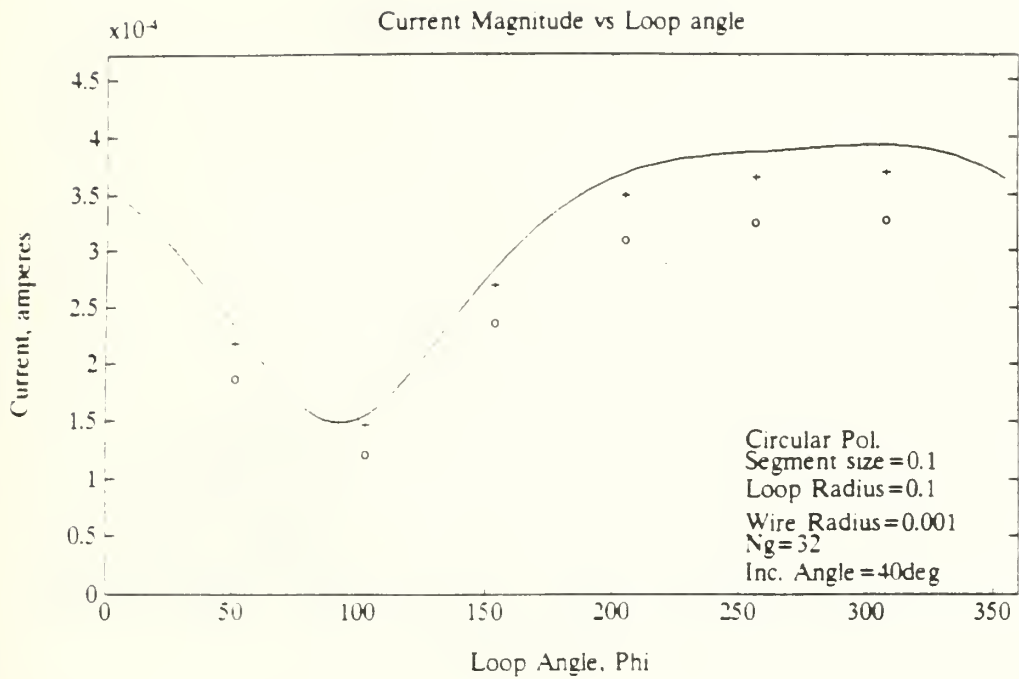


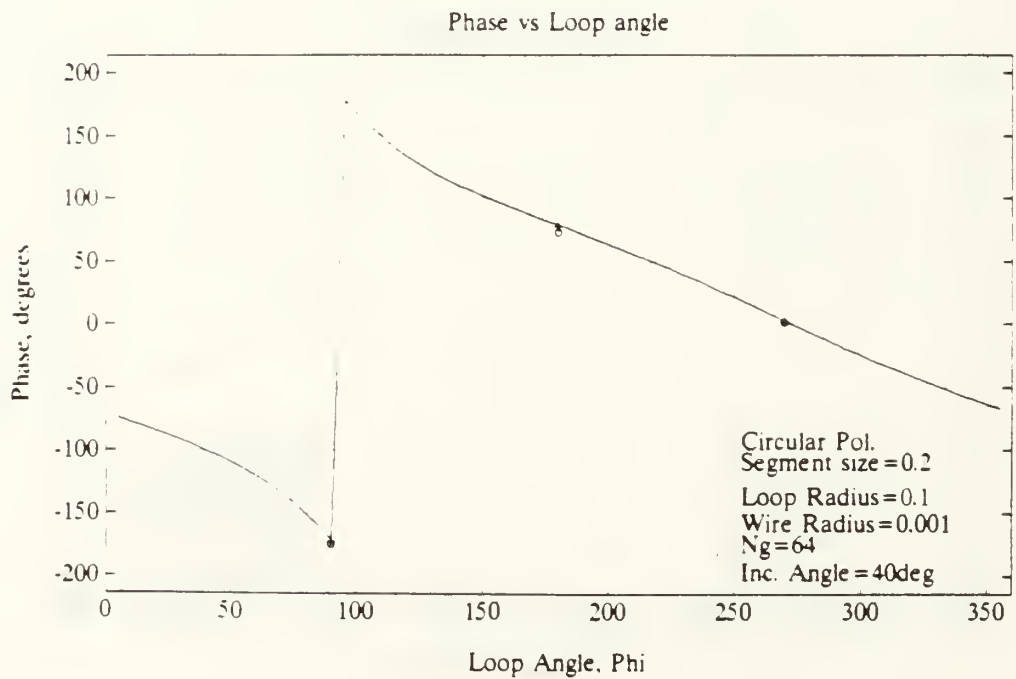
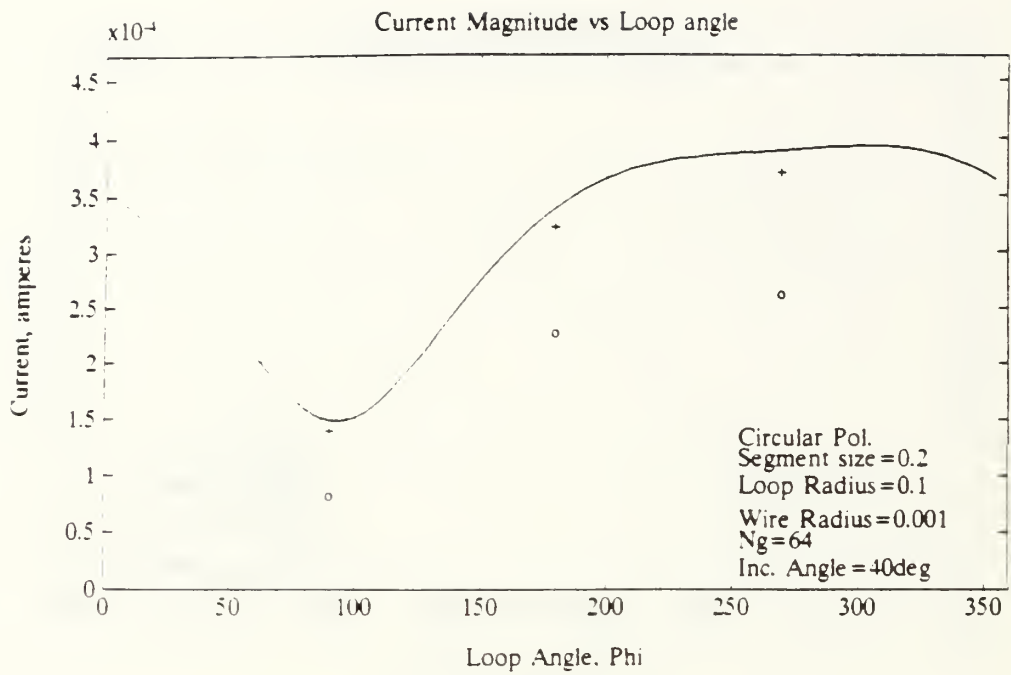


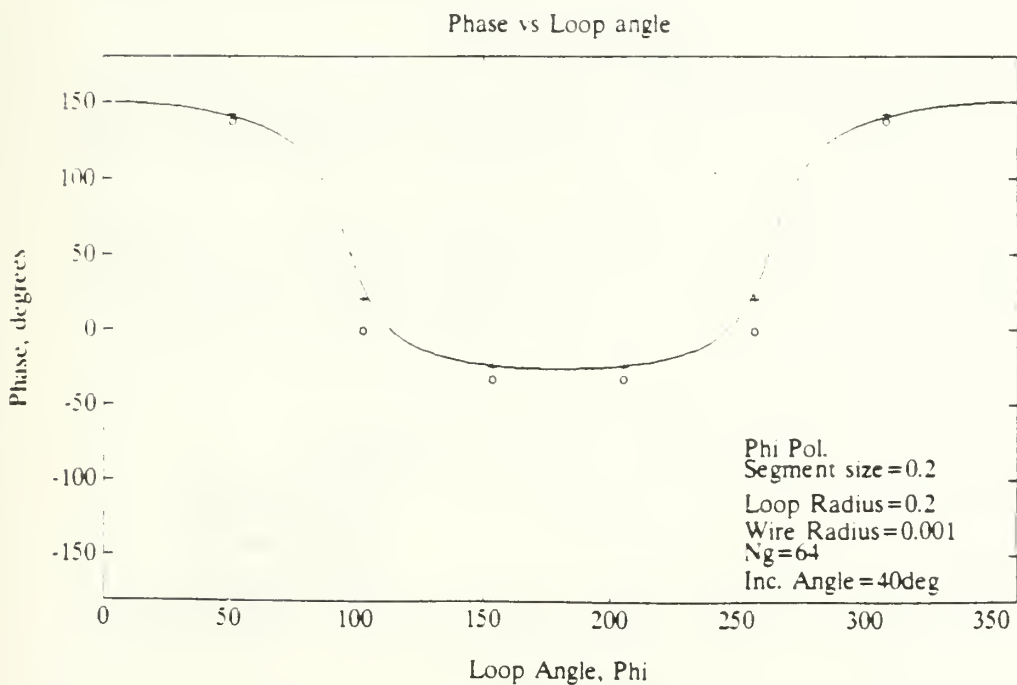
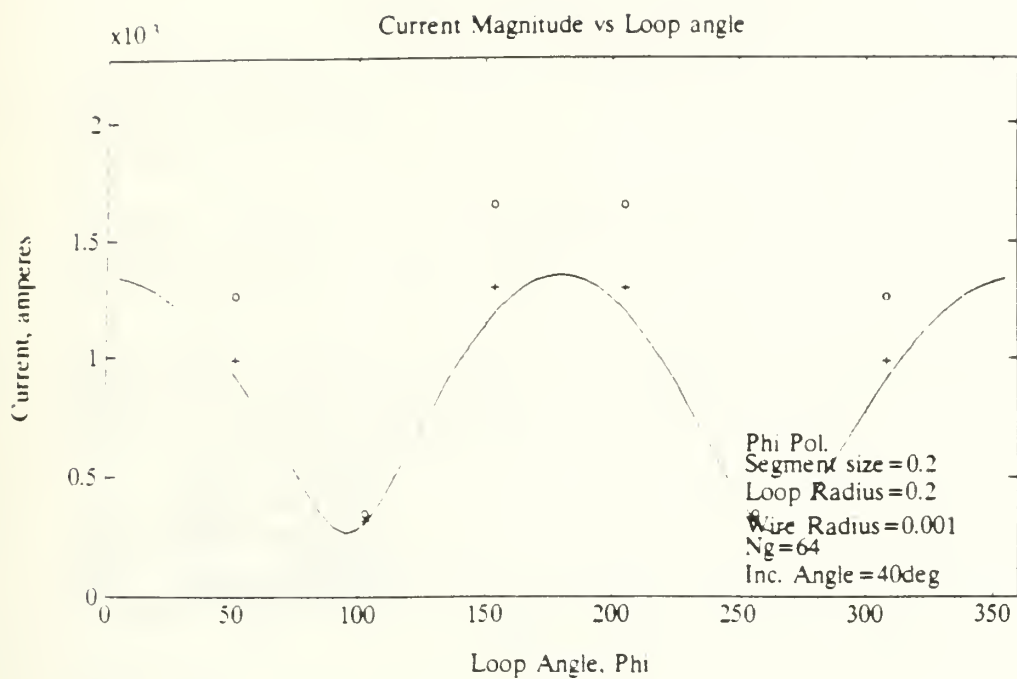


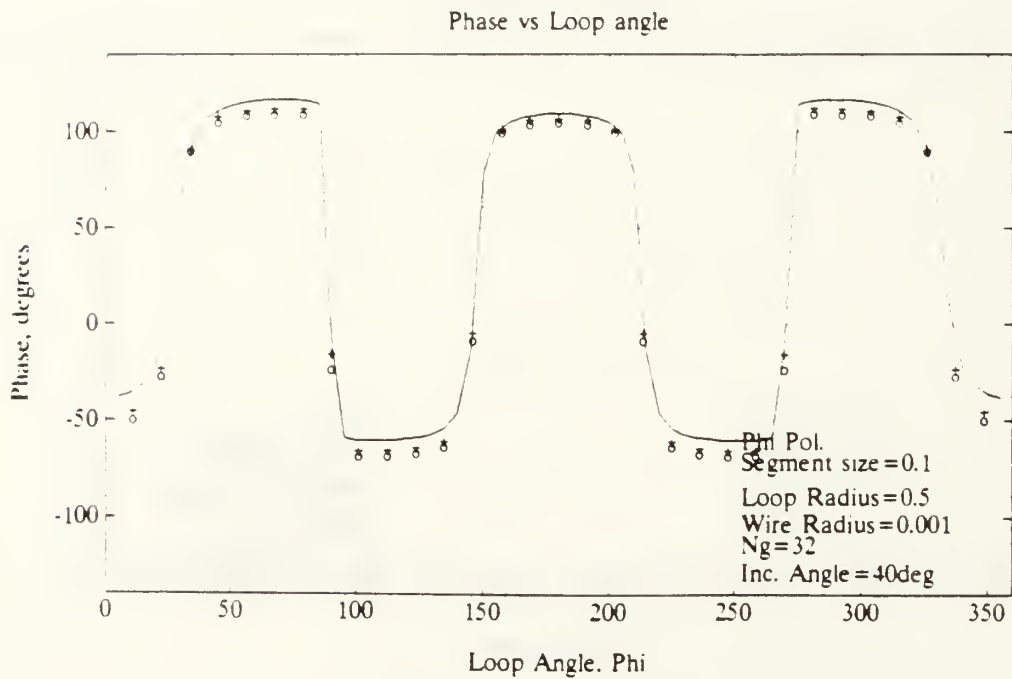
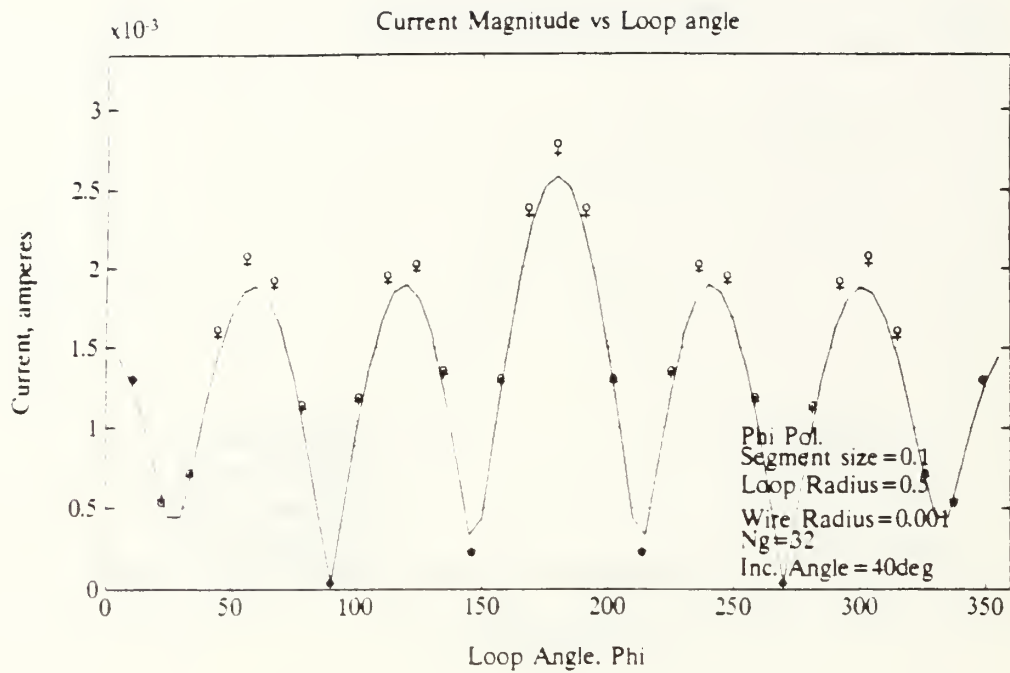












INITIAL DISTRIBUTION LIST

- | | | |
|----|---|---|
| 1. | Library, Code 52
Naval Postgraduate School
Monterey, CA 93943-5002 | 2 |
| 2. | Defense Technical Information Center
Cameron Station
Alexandria, VA 22304-6145 | 2 |
| 3. | Chairman, Code EC
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA 93943-5002 | 1 |
| 4. | Commander
Naval Air Test Center
SY90 Mr. L. Krouse
Patuxent River, MD 20670-5304 | 1 |
| 5. | Commander
Naval Air Test Center
SY95BW Mr. B. Walter
Patuxent River, MD 20670-5304 | 1 |
| 6. | Professor D. C. Jenn, Code EC/Jn
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA 93943-5002 | 1 |
| 7. | Michael A. Morgan, Code EC/Mw
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA 93943-5002 | 1 |

Thesis
W22415 Walter
c.1

The use of conformal
subdomain basis functions
in the method of moments
computations for a thin
wire.

Thesis
W22415 Walter
c.1

The use of conformal
subdomain basis functions
in the method of moments
computations for a thin
wire.

DUDLEY KNOX LIBRARY



3 2768 00035907 9